STUDIES ON SOLUTION METHODS FOR NONLINEAR PROGRAMMING PROBLEMS

Masao FUKUSHIMA

January 1979

STUDIES ON SOLUTION METHODS

FOR

NONLINEAR PROGRAMMING PROBLEMS

MASAO FUKUSHIMA

KYOTO UNIVERSITY

KYOTO, JAPAN

JANUARY 1979

STUDIES ON SOLUTION METHODS

FOR

NONLINEAR PROGRAMMING PROBLEMS

by

Masao Fukushima

Submitted in partial fulfillment of the

requirement for the degree of

DOCTOR OF ENGINEERING

at

Kyoto University

KYOTO, JAPAN

January 1979

PREFACE

Optimization is a principle which plays an important role in various fields of engineering. In practice, an optimization problem is normally formulated as a mathematical model consisting of an objective function to optimize and constraints specifying feasible solutions. Such a problem is generally called a mathematical programming problem. The analysis of mathematical programming problems containing, in particular, nonlinearity in the objective function or the constraints is called nonlinear programming. Specifically, nonlinear programming involves such diverse fields as characterizing optimality conditions , clarifying duality correspondences and developing numerical methods to solve such problems.

The origin of nonlinear programming goes as far back as the pioneering work of Kuhn and Tucker in 1951. Most attention in the early years was mainly focused on the theoretical subjects like optimality and duality theories. Later, rapid progress of computer machineries in the 1960's has made it possible to handle relatively large and complex problems, and hence the development of efficient numerical methods has become a major subject of research in this field. In those years, however, most of the solution techniques were proposed independently and the interrelationship between theory and practice remained almost neglected. Only recently, it has been recognized that theory and practice should be integrated to provide a unified framework for the study of nonlinear programming. Moreover, one of the recent trends is to study the problems under a still more general setting and to develop effective numerical methods applicable to such problems by establishing a theoretical framework based on new concepts. On the other hand, it is also necessary to restrict our attention to a certain class of problems and then develop a detailed theory and a most relevant numerical solution technique.

With these situations in mind, this thesis attempts to present new ideas and to give some insights into the nonlinear programming problems. Nonlinear programming is a relatively young field of study and numerous attempts are now being undertaken extensively. It seems that nonlinear programming is still less established in both theory and practice compared with linear programming. The author hopes that the work contained in the thesis will be helpful for future research in this growing field.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Professor H. Mine of Kyoto University for his invaluable guidance and criticism ever since my student days. Without his constant encouragement and helpful suggestions, this work would not exist today.

I also wish to acknowledge Associate Professor K. Ohno of Kyoto University for his inenumerable suggestions and comments. Particularly, the work in Chapters 7 through 9 of this thesis was motivated from discussions with him.

I appreciate the stimulating atmosphere supplied by the Department of Applied Mathematics and Physics at Kyoto University. In this respect, I wish to acknowledge Professor T. Hasegawa, Associate Professor T. Ibaraki, Dr. H. Kawai, and others, too many for me to enumerate individually. Furthermore, I am grateful to all my friends and colleagues who have encouraged and assisted me toward the completion of the thesis. In particular, I have to mention that Mr. Y. Fujii has partly participated in completing the work in Chapter 4 of the thesis. Finally, I wish to express my appreciation to Mr. N. Navarrete, Jr. for his kindness of checking and correcting my English.

CONTENTS

CHAPTER 1

INTRODUCTION

This chapter states the historical background and the present status of nonlinear programming and present the outline of this thesis.

1.1 Nonlinear Programming

The mathematical programming problem is that of minimizing (or maximizing) a certain objective function subject to constraints with respect to decision variables. It is frequently encountered not only in operations research but also various fields such as economics, business, engineering, and other social and natural sciences. If the objective function and the constraint functions are not all linear, the problem is called the *nonlinear programming (NLP) problem* in contrast with the *linear programming (LP) problem* in which all functions are linear. Moreover, if the objective function and the constraints are both convex, then the problem is called the *convex programming problem* and is of particular importance due to many nice properties it has.

The area of NLP covers a variety of topics such as devising methods of finding the optimal solutions, investigating the algorithmic characteristics of those methods, and studying various types of optimality conditions and duality correspondences. The

NLP has a history of about thirty years since it became commonly recognized as a separate field in operations research. There still remains, however, a number of subjects not yet resolved, while LP has almost completely been established in theory via comprehensive studies of systems of linear equalities and inequalities and has extensively been applied in practice as a powerful tool of decision making.

Nevertheless, starting from the famous pioneering work on NLP by Kuhn and Tucker and passing through the theory of convex analysis of Fenchel and Rockafellar who succeeded in outgrowing classical analysis by systematically replacing differentiability assumptions by convexity assumptions, a number of important theoretical results have been obtained up to the present. For convex programming, in particular, excellent unified theory has been established which stands comparison with theory for LP.

On the other hand, as to computational aspects of finding optimal solutions of NLP problems, several general purpose algorithms have been devised in proportion to the progress of computer machineries in recent years. Those algorithms are, however, still unsatisfactory from a practical viewpoint, and continual efforts are being made to improve the efficiency and to overcome various difficulties in solving actual problems. It seems to the author that research in this branch will go on boundlessly since it is unlikely to win an absolutely superior method such as the simplex methods in LP.

1.2 Review of NLP Theory

Kuhn and Tucker succeeded in extending the classical Lagrange multiplier rule for equality constrained NLP problems to the case of inequalities in their fundamental paper [K6] in 1951. Prior to Kuhn and Tucker, Fritz John [J2] had made an attempt to generalize the method of Lagrange to inequality constrained problems, but his result was unsatisfactory in the sense that certain irregular situations could not be excluded. Kuhn and Tucker pointed out that certain regularity condition, called constraint qualification, need be considered to avoid such difficulties. The optimality conditions obtained by them have been called the Kuhn-Tucker conditions after their names, though it was found later that Karush [K2] had derived the equivalent conditions prior to them. It is very interesting to note that the motivations for the works by Kuhn and Tucker, John, and Karush were entirely different from each other. Namely, Karush was inspired from the calculus of variations, John considered a problem in geometrical inequality, and the background of Kuhn and Tucker was in a wide variety of fields such as network theory, duality in topology, the theory of games and linear programming. Therefore, it may be reasonable to say that the present research of NLP has its origin at the work of Kuhn and Tucker. In fact, it was their paper that gave it the name nonlinear programming.

Since then, study of the optimality conditions has been one of the major topics in NLP. In particular, a number of results were published during the 1960's and several of them were collected by Fiacco and McCormick [F8] and Mangasarian [M1]. Moreover, constraint qualifications other than that of Kuhn and Tucker have been proposed and more sophisticated optimality conditions have been obtained by many authors. Relationships among various constraint qualifications were clarified by Bazaraa, Goode and Shetty [B1].

Recently, attempts have begun to extend the Kuhn-Tucker conditions to more general NLP problems involving nondifferentiable functions. In particular, Pshenichnyi [P8] in 1971 considered necessary optimality conditions for the NLP problem with quasi-differentiable functions, and Clarke [C4] in 1976 and Hiriart-Urruty [H4] in 1978 studied the NLP problems involving functions which have generalized gradients in the sense of Clarke [C3].

Kuhn and Tucker [K6] also showed the equivalence of an NLP problem and a saddle point problem for the Lagrangean associated with the original problem. In the late 1960's, Gale [G1] and Geoffrion [G5] developed an elegant duality theory for convex programming by the use of a perturbation function. Duality theory in convex programming has further been established almost completely during a series of works of Rockafellar whose idea is to utilize the concept of conjugate functions originated by Fenchel [F2]. Most of the results of Rockafellar may be found in [R5][R7].

For nonconvex programs, an earlier attempt was made by Gould [G9] in 1969 to generalize the Kuhn-Tucker equivalence theorem concerning the saddle point theorem by introducing the notion of multiplier functions instead of Lagrange multipliers. The multiplier functions were further investigated in detail and the conditions for the local saddle point theorem to hold were studied by Arrow, Gould and Howe [A6].

On the other hand, by utilizing nonlinear perturbation functions, Rockafellar showed in his recent papers [R9][R10] that many results already obtained with respect to convex programs could be extended even for nonconvex cases, and he constructed an elegant nonconvex duality theory. These results are particularly important since they establish a background for the multiplier methods for solving general NLP problems.

1.3 Review of NLP Methods

Methods of solving NLP problems may be roughly classified into two categories, i.e., primal methods and dual methods. Typical primal methods in early years are feasible direction methods by Zoutenzijk [Z4] and the gradient projection method by Rosen [R12]. The latter, combined with the variable metric methods for unconstrained minimization, has been improved by Goldfarb [G7] and Murtagh and Sargent [M19] to assure fast convergence.

Generalizing the simplex method of LP, Wolfe [W5] proposed the reduced gradient method for solving NLP problems with linear constraints. Later this method has been modified by Abadie [A1] [A2] to cope with nonlinear constraints. This generalized reduced gradient (GRG) method is now recognized as one of the most efficient methods in NLP.

More recently, primal methods that require a solution of an auxiliary quadratic programming problem at each iteration have been proposed and shown to converge rapidly by Han [H1] in 1976 and Powell [P7] in 1977.

Dual methods of primitive type are the penalty methods which are often called sequential unconstrained minimization techniques (SUMT), compiled by Fiacco and McCormick [F8], although they seem not to have been explicitly recognized as dual methods. In the 1960's, penalty methods were paid much attention from practical viewpoints due to their simplicity and wide applicability to actual problems. However, it has been gradually recognized that penalty methods had a serious difficulty in the sense that the penalty functions tend to be considerably ill-conditioned as the iteration proceeds. These properties of penalty methods have been studied in detail by Lootzma [L5] and Ryan [R14].

In order to avoid such a difficulty, new methods, which make use of penalty functions combined with the Lagrangean, were proposed in 1969 independently by Hestenes [H2] and Powell [P6]. These methods have indeed clear dual characteristics and are called the multiplier methods. It should be noted, however, that the origin of their idea may be found in an earlier paper by Arrow and Solow [A7]. Although Hestenes and Powell considered only equality constrained problems, their idea has later been generalized to deal with inequality constraints as well by Rockafellar [R6] and Kort and Bertsekas [K5]. The multiplier methods are now widely accepted as one of the superior methods for NLP problems. The efficiency of the multiplier methods has been validated by Bertsekas [B7][B10]. Furthermore, interesting approaches based on the multiplier methods have been proposed by Mangasarian [M2], Fletcher [F10] and Tapia [T1].

Very recently, nondifferentiable optimization problems have drawn significant attention since actual problems often involve functions not everywhere differentiable. Earlier attempts were made for problems with nondifferentiable convex functions by Bertsekas and Mitter [B11] in 1973, Wolfe [W6] in 1975 and Lemarechal [L2] in 1975. Subsequently, the idea of Wolfe and

Lemarechal has been generalized by Feuer [F4] in 1976 and Mifflin [M4] in 1977 to handle more general classes of non-differentiable functions. Nondifferentiable optimization is one of the newest topics in NLP and is now under active investigation.

In addition to the above mentioned methods for general NLP problems, there are also many algorithms devised for large-scale mathematical programming problems. Most of the earlier works in this field were to deal with large-scale LP problems. Among them, the most important result might be the Dantzig-Wolfe decomposition principle [D1]. For mathematical programming problems other than LP problems, there have been proposed several methods such as Benders' decomposition methods [B4][G6], Rosen's partitioning procedures [R13] and Geoffrion-Silverman's primal decomposition methods [G4][S3]. These methods for large-scale problems are collected in Lasdon [L1] and Geoffrion [G3]. On the other hand, based on the principle of optimality in dynamic programming (DP) by Bellman [B2], several attempts have been made to deal with large-scale mathematical programming problems. It seems, however, that DP has been relatively less applied to large-scale NLP problems compared with the above mentioned methods for large-scale problems. Thus it might be valuable to study the applicability of DP to various classes of large-scale NLP problems.

1.4 Outline of the Thesis

The main purpose of this thesis is to clarify various properties of NLP problems and to propose several methods to solve these problems. First, a foundation of traditional SUMT for convex programming problems is studied, and a dual method of SUMT is proposed to overcome computational difficulties in the traditional SUMT. The same idea as that used in the dual method is also applied to multifacility location problems. Next, certain nonconvex programming problems are also considered and algorithms for solving these problems are proposed. . . . Finally, applicability of DP to various NLP problems is investigated in detail. Organization of each chapter is briefly summarized as follows:

In Chapter 2, a class of penalty functions for solving convex programming problems with general constraint set is considered. Convergence theorems for penalty methods are established by utilizing the notion of infimal convergence of a sequence of functions. It is shown that most existing penalty functions are included in the class of penalty functions defined in this chapter. Thus this chapter provides a foundation of penalty methods for convex programming problems.

In Chapter 3, a new class of sequential unconstrained optimization methods, called the conjugate penalty methods, is proposed for solving convex programming problems. The conjugate penalty methods utilize conjugate convex functions and are based on Fenchel's duality theorem in convex analysis. It is shown that, under certain assumptions, conjugate penalty functions behave quite mildly and hence avoid the ill-conditioning of ordinary penalty methods of Chapter 2. Convergence of the methods is proved and the relationship between ordinary and conjugate penalty methods is shown.

In Chapter 4, a dual approach is developed for solving multifacility location problems under very general settings. Formulation of the dual problem is essentially based on Fenchel's duality theorem which also plays a crucial role in the previous chapter. It should be noted that the multifacility location problem is applicable to various actual problems, for example, locating facilities such as plants, warehouses and firestations, locating machines in a factory, and communication networks.

In Chapters 5 and 6, two algorithms are proposed to minimize a general function which is the sum of a continuously differentiable function and a convex function. The objective function is thus in general neither convex nor differentiable. It is noted that this class of problems contains, as a special case, the problem of minimizing a continuously differentiable function over a closed convex set. In view of this, the algorithm proposed in Chapter 5 is regarded as a natural extension of the Frank-Wolfe method to the more general problems. On the other hand, the algorithm proposed in Chapter 6 is a generalized version of the proximal point algorithm for monotone operators. Convergence of the algorithms is proved and the rate of convergence is discussed.

In Chapter 7 through 9, sufficient conditions are presented under which large-scale NLP problems are decomposed into subproblems by dynamic programming. Specifically, Chapter 7 considers multi-level decomposition of general NLP problems. Chapter 8 deals with Stochastic NLP problems with chance-constraints. Chapter 9 is concerned with multiple criteria NLP problems for which the concept of Pareto optimality replaces the ordinary concept of optimality. The underlying ideas in Chapters 7 through 9 are basically the same, although methods used in proving the results are significantly different from each other.

The final chapter is conclusion of the thesis. Appendix provides a summary of definitions and notations in convex analysis that are frequently used in the thesis.

Finally, it should be mentioned that the materials of the thesis are based on the author's original works. Specifically, Chapter 2 is taken from the paper [M7] published in Journal of Optimization Theory and Applications, Chapter 3 from the papers [M15] and [M6] published, respectively, in SIAM Journal on Control and Optimization and in Memoirs of the Faculty of Engineering, Kyoto University, Chapter 4 from the paper [M10] published in the Proceedings of the International Conference on Cybernetics and Society, Tokyo, 1978, Chapter 5 from the paper [M8] to be published in Journal of Optimization Theory and Applications, Chapter 6 from the paper [F14] to be presented at the Tenth International Symposium on Mathematical Programming, Montreal, 1979, Chapters 7 and 8 from the papers [M13] and [M14], respectively, published in Journal of Mathematical Analysis and Applications, and Chapter 9 from the paper [M9] to be published in the International Journal of Systems Science.

CHAPTER 2

FOUNDATION OF PENALTY FUNCTION METHODS FOR GENERAL CONVEX PROGRAMS

In this chapter, a class of penalty functions for solving convex programming problems with general constraint sets is considered. Convergence theorems for the penalty function methods are established by utilizing the concept of infimal convergence of a sequence of functions. It is shown that most existing penalty functions are included in the class of penalty functions discussed in this chapter.

2.1 Introduction

We consider the following general convex programming problem:

(P) minimize $f(x)$ subject to $x \in C$,

where f is a closed convex function on R^n and C is a closed convex subset of R^n . Many kinds of sequential unconstrained minimization methods for solving problem (P) have been appeared in the literature. Those methods are surveyed in, for example, [F8] [L5]. A typical idea underlying those techniques is transforming problem (P) into a sequence of unconstrained problems, say $\{(P_k);\ k=1,2,\dots\}$, by means of a sequence of auxiliary functions which in general contain one or several parameters. We shall term those functions as *penalty* *functions* conventionally. The constraint set C of problem (P) is usually represented by the system of equalities and/or inequalities of functions and those auxiliary functions are composed in terms of the problem functions. In the following development, however, we do not explicitly assume the constraint functions.

The purpose of this chapter is to develop a unified theory of sequential unconstrained minimization methods for the general convex programming problem (P). We shall consider penalty functions for (P) without taking account of constraint functions. It is noted that Fiacco [F5][F6] and Fiacco and Jones [F7] generalize and characterize the penalty function method for solving (P), where they do not assume convexity. However, their approaches are essentially based on the interior and exterior penalty function theory. On the other hand, we consider here only convex cases, but our characterization is more general than those, because we make no distinction either between the interior and the exterior penalty functions or between equality and inequality constraints.

This chapter is organized as follows: Section 2.2 describes the general penalty function methods. Section 2.3 defines the concept of infimal convergence of functions that plays a crucial role in this chapter. Section 2.4 proves the convergence theorems for the penalty function methods. Section 2.5 shows that the results given here are applicable to most existing penalty function methods.

2.2 A Penalty Function Approach

It is obvious that problem (P) is equivalent to the following problem;

(P) minimize $f(x) + \delta_C(x)$ over $x \in R^n$,

where δ_C is the *indicator function* of C defined by

$$\delta_C(x) = 0 \text{ if } x \in C \text{ , } = +\infty \text{ if } x \notin C \text{ .}$$

Since C is closed and convex, δ_C is a closed convex function.

In the following, we shall assume that the minimum set, denoted by M , is nonempty and bounded, hence

$$M = \{ x \in C \text{ ; } f(x) \leqq f(y) \quad \text{for all} \quad y \in C \}$$

is a compact convex set. This assumption is the same as assuming that f and C have no direction of recession in common [R5 , Thm.27.3], or equivalently that $f + \delta_C$ has no direction of recession [R5 , Thm.27.2]. The reader may refer to [A3] for existence criteria of minima. We shall make an additional assumption that $\text{int}(\text{dom } f) \supset C$, which will be needed in the proof of the theorem given in Section 2.4. However, there will be little loss of generality in making this assumption, because in most practical problems f is everywhere finite, i.e., $\text{dom } f = R^n$.

In solving problem (P) via a penalty method, it is required in general thet

A1. For a sequence of penalty functions $\{ U_k \text{ ; } k=1,2,\dots \}$, there is a sequence $\{ x_k \text{ ; } k=1,2,\dots \}$ where each x_k minimizes $U_k(x)$ over R^n , i.e., $U_k(x_k) = \min_{x \in R^n} U_k(x)$.

A2. Every cluster point of $\{x_k ; k=1,2,\ldots\}$ is in M and $U_k(x_k)$ tends to the minimum value of (P).

Therefore, a feature of penalty function algorithms is successive minimization of the following type:

(P_k) minimize $U_k(x)$ over $x \in R^n$, $k=1,2,\ldots$.

Let us suppose that the penalty functions U_k take the form

$$U_k(x) = f(x) + h_k(x)$$

as most penalty functions do. In order to fulfill the requirements A1 and A2, it is expected that the whole information of the constraint set C of problem (P) is imbedded in the sequence of functions $\{h_k ; k=1,2,\ldots\}$, or intuitively speaking, that h_k converges to δ_C in a certain sense.

2.3 Infimal Convergence

In this section, we sketch the notion of infimal convergence of a sequence of functions to a function, introduced by Wijsman [W2][W3]. For any function g on R^n and for any $r > 0$, define the function ${}_rg$ on R^n by

$$ {}_rg(x) = \inf_{y \in B_r(x)} g(y) , $$

where $B_r(x)$ is the closed sphere of radius r with center x.

<u>Definition 2.1</u>. Let $\{ g_k ; k=1,2,\ldots \}$ be a sequence of functions on R^n. The sequence is said to *converge infimally* to a function g if for every $x \in R^n$

$$ \lim_{r\to 0} \liminf_{k\to\infty} {}_rg_k(x) = \lim_{r\to 0} \limsup_{k\to\infty} {}_rg_k(x) = g(x) . $$

If $\{ g_k \}$ converges infimally to g, we write $g_k \xrightarrow[\inf]{} g$, while for pointwise convergence, we write as usual $g_k \longrightarrow g$. It is noted as indicated by Wijsman [W3] that infimal and pointwise convergence are not comparable in general. But it is intuitively clear that if $g_k \longrightarrow \bar{g}$ and $g_k \xrightarrow[\inf]{} g$, then $\bar{g} \geqq g$.

<u>Example 2.1</u>. For $k=1,2,\ldots$, let $g_k : R \longrightarrow (-\infty,+\infty]$ be

$$ g_k(x) = kx \quad \text{if} \quad -1/k \leqq x \leqq 1/k , \quad = +\infty \quad \text{otherwise.} $$

Then

$$ g_k \longrightarrow \bar{g} , \text{ where } \bar{g}(x) = 0 \quad \text{if} \quad x = 0 , \quad = +\infty \quad \text{if} \quad x \neq 0 $$

and

$$ g_k \xrightarrow[\inf]{} g , \text{ where } g(x) = -1 \quad \text{if} \quad x = 0 , \quad = +\infty \quad \text{if} \quad x \neq 0 . \quad . \quad . $$

2.4 Convergence Theorems

In this section, we consider a class of penalty functions which guarantee the validity of a sequential unconstrained minimization method for solving (P) and then for an important subclass of (P) we restate it in a somewhat concrete form.

In order that property A1 holds, we shall always assume the following condition:

B1. For every k, h_k is a closed convex function with $\mathrm{dom}\, f \cap \mathrm{dom}\, h_k \neq \emptyset$ and f and h_k have no direction of recession in common.

Under this condition, the minimum set of (P_k), denoted by M_k, is nonempty, compact and convex.

The key condition for property A2 to hold is the following:

B2. $\{ h_k \ ; \ k=1,2,\ldots \}$ converges infimally to δ_C.

The following theorem asserts that a sequential unconstrained minimization method via penalty functions which satisfy B1 and B2 is valid for solving (P).

Theorem 2.1. Under conditions B1 and B2, every sequence $\{ x_k \in M_k \ ; \ k=1,2,\ldots \}$ has a convergent subsequence and each cluster point of the sequence is in M. Moreover, $\lim_{k\to\infty} U_k(x_k) = \min_C f$.[†]

Proof. First we prove that the sequence $\{ U_k \}$ converges infimally to $f + \delta_C$. Suppose $x \in C$. As $C \subset \mathrm{int}(\mathrm{dom}\, f)$ by our assump-

---

† $\min_C f = \min_{x\in C} f(x)$.

tion, there exists $r_0 > 0$ such that $B_r(x) \subset \mathrm{int}(\mathrm{dom}\, f)$ if $0 < r < r_0$. By definition, ${}_rU_k(x) = \inf_{y \in B_r(x)} \{ f(y) + h_k(y) \}$. Since f is convex, f is continuous on $\mathrm{int}(\mathrm{dom}\, f)$, that is, $y \in B_r(x)$ implies $f(x) - \varepsilon < f(y) < f(x) + \varepsilon$ and $\varepsilon \to 0$ as $r \to 0$. Thus we have

$$ {}_rU_k(x) < \inf_{y \in B_r(x)} \{ f(x) + \varepsilon + h_k(y) \} = f(x) + \varepsilon + {}_rh_k(x) $$

and

$$ {}_rU_k(x) > \inf_{y \in B_r(x)} \{ f(x) - \varepsilon + h_k(y) \} = f(x) - \varepsilon + {}_rh_k(x) , $$

hence

$$ \lim_{r\to 0} \limsup_{k\to\infty} {}_rU_k(x) \leqq \lim_{r\to 0} \{ f(x) + \varepsilon + \limsup_{k\to\infty} {}_rh_k(x) \} = f(x) $$

and

$$ \lim_{r\to 0} \liminf_{k\to\infty} {}_rU_k(x) \geqq \lim_{r\to 0} \{ f(x) - \varepsilon + \liminf_{k\to\infty} {}_rh_k(x) \} = f(x) . $$

For $x \notin C$, it is obvious that

$$ \lim_{r\to 0} \limsup_{k\to\infty} {}_rU_k(x) = \lim_{r\to 0} \liminf_{k\to\infty} {}_rU_k(x) = +\infty . $$

Consequently

$$ \lim_{r\to 0} \limsup_{k\to\infty} {}_rU_k = \lim_{r\to 0} \liminf_{k\to\infty} {}_rU_k = f + \delta_C . $$

Choose $\varepsilon > 0$ arbitrarily. Since U_k has no direction of recession, the set $M_k^\varepsilon = \{ x ; U_k(x) \leqq \min_C f + \varepsilon \}$ is compact for every k. Similarly, the set $M^\varepsilon = \{ x \in C ; f(x) \leqq \min_C f + \varepsilon \}$ is compact. Since U_k converges infimally to $f + \delta_C$, applying [R5 , Thm.7.1] and taking account of compactness, M_k^ε converges to M^ε with respect to the Hausdorff metric as $k \to \infty$. (Definition of the Hausdorff metric on a space of compact subsets is found, for example, in Berge [B6].) Since M_k^ε is nonempty for every k greater than some K, $M_k \subset M_k^\varepsilon$ for $k \geqq K$. Therefore every sequence $\{x_k \in M_k\}$ has a convergent subsequence and each cluster point is in M^ε. On the other hand, M^ε is nonempty and compact, and M^ε converges nonincreasingly to M with respect to the Hausdorff metric as ε tends to zero [R5 , Thm.27.2]. Therefore every cluster point of $\{x_k\}$ is in M.

Now we prove the last assertion. Without loss of generality, we assume that $\{x_k\}$ converges to $\bar{x} \in M$. Then

$$\begin{aligned}
f(\bar{x}) + \delta_C(\bar{x}) &= \lim_{r\to 0} \limsup_{k\to\infty} {}_rU_k(\bar{x}) \\
&\geqq \limsup_{k\to\infty} U_k(x_k) \qquad (\text{ since } x_k \text{ minimizes } U_k) \\
&\geqq \liminf_{k\to\infty} U_k(x_k) \\
&\geqq \lim_{r\to 0} \liminf_{k\to\infty} {}_rU_k(\bar{x}) \qquad (\text{ by the definition of } {}_rU_k) \\
&= f(\bar{x}) + \delta_C(\bar{x}) ,
\end{aligned}$$

since U_k converges infimally to $f + \delta_C$. Therefore

$$\lim_{k\to\infty} U_k(x_k) = \min_C f .$$

This completes the proof. □

As indicated in Section 2.3, pointwise convergence does not imply infimal convergence. In practice, it seems difficult to establish the infimal convergence condition B2. However, if C has nonempty interior, it will be shown that there is a close relationship between pointwise and infimal convergence. If C is represented in terms of the system of inequalities of functions, then int $C \neq \emptyset$ is ordinarily satisfied except for pathological cases. The following condition will be substituted for B2:

B2'. $\{h_k\}$ converges pointwise to δ_C except on the boundary of C, i.e., $\lim_{k\to\infty} h_k(x) = 0$ if $x \in \text{int } C$, $= +\infty$ if $x \notin C$.

It should be noted that this condition says nothing explicitly about the behavior of $\{h_k\}$ on the boundary of C. The following lemma, however, shows that, for any boundary point x of C, $h_k(x)$ never tends to a negative limit.

Lemma 2.1. Assume that int C is nonempty. If $\{h_k\}$ is a sequence of convex functions satisfying B2', then

$$\liminf_{k\to\infty} h_k(x) \geqq 0$$

for any x on the boundary of C.

Proof. Let x by any boundary point of C. There exist x_1, $x_2 \in \text{int } C$ such that $x_1 \neq x_2$ and $x_1 = \lambda x + (1-\lambda)x_2$ for some $0 < \lambda < 1$. By the convexity of h_k, for every k

$$h_k(x) \geqq \{ h_k(x_1) - (1-\lambda)h_k(x_2) \} / \lambda .$$

Thus

$$\liminf_{k\to\infty} h_k(x) \geqq \lim_{k\to\infty} \{ h_k(x_1) - (1-\lambda)h_k(x_2) \} / \lambda$$
$$= 0 . \qquad \square$$

Theorem 2.2. Assume that int C is nonempty. If $\{h_k\}$ is a sequence of convex functions satisfying B2', then the sequence converges infimally to δ_C.

Proof. By definition, it suffices to show that

$$\lim_{r\to 0} \liminf_{k\to\infty} {}_rh_k(x) \geqq 0 \quad \text{if} \quad x \in C , \quad = +\infty \quad \text{if} \quad x \notin C$$

and

$$\lim_{r\to 0} \limsup_{k\to\infty} {}_rh_k(x) \leqq 0 \quad \text{if} \quad x \in C .$$

First we prove that $\lim_{r\to 0} \liminf_{k\to\infty} {}_rh_k(x) \geqq 0$ for all $x \in R^n$. Since $\operatorname{int} C$ is nonempty, there is an n-dimensional simplex S with vertices $\{ b_0, b_1, \ldots, b_n \}$ in $\operatorname{int} C$. For any $\varepsilon > 0$, there exist K_i, $i=0,1,\ldots,n$, such that $|h_k(b_i)| < \varepsilon$ for all $k \geqq K_i$, $i=0,1,\ldots,n$. Put $K = \max\{K_0, K_1, \ldots, K_n\}$. Let $\bar{b}$ be the barycenter of S. For any k, epi h_k is supported by a hyperplane, say $z = \langle a_k, x\rangle + \beta_k$, at $\bar{b}$. In particular, for $k \geqq K$ we have

$$-\varepsilon < h_k(\bar{b}) = \langle a_k, \bar{b}\rangle + \beta_k < \varepsilon$$

and

$$\varepsilon > h_k(b_i) \geqq \langle a_k, b_i\rangle + \beta_k , \qquad i=0,1,\ldots,n .$$

That is, $\langle a_k, b_i - \bar{b}\rangle < 2\varepsilon$, $i=0,1,\ldots,n$. Since ε is arbitrary, a_k tends to a null vector and β_k also tends to zero as $k \to \infty$. Therefore

$$\begin{aligned}
\lim_{r\to 0} \liminf_{k\to\infty} {}_rh_k(x) &= \lim_{r\to 0} \liminf_{k\to\infty} \inf_{y\in B_r(x)} h_k(y) \\
&\geqq \lim_{r\to 0} \liminf_{k\to\infty} \inf_{y\in B_r(x)} \{ \langle a_k, y\rangle + \beta_k \} \\
&= \lim_{r\to 0} \liminf_{k\to\infty} \{ \langle a_k, x\rangle - r\|a_k\| + \beta_k \} \\
&= 0 .
\end{aligned}$$

Secondly we prove that for $x \notin C$, $\lim_{r\to 0} \liminf_{k\to\infty} {}_rh_k(x) = +\infty$.

Assume the contrary. Then we must have a sequence $\{x_k\}$ converging to x and $\lim_{k\to\infty} h_k(x^k) < +\infty$. For such a sequence, we can choose $y \notin C$ and generate a sequence $\{z^k\}$ in int C such that $y = (x^k + z^k)/2$ and $z^k \to z \in \text{int } C$. The existence of such y follows from the closedness of C. For sufficiently large k, we may assume that every z^k stays in an n-dimensional simplex S in int C. As shown previously, for any $\varepsilon > 0$ there exists some K such that $h_k(b_i) < \varepsilon$ for all vertices b_i of S. Since h_k is convex and each $z' \in S$ is represented as $z' = \lambda_0 b_0 + \lambda_1 b_1 + \ldots + \lambda_n b_n$, where $\lambda_0 + \lambda_1 + \ldots + \lambda_n = 1$ and each $\lambda_i \geqq 0$,

$$h_k(z') \leqq \lambda_0 h_k(b_0) + \lambda_1 h_k(b_1) + \ldots + \lambda_n h_k(b_n) < \varepsilon$$

for $k \geqq K$. Thus $\limsup_{k\to\infty} h_k(z^k) \leqq 0$. Again by the convexity of h_k, $h_k(x^k) \geqq 2h_k(y) - h_k(z^k)$. Consequently $\lim_{k\to\infty} h_k(x^k) = +\infty$, because $\liminf_{k\to\infty} \{ 2h_k(y) - h_k(z^k) \} = +\infty$. This is a contradiction.

Lastly we show that $\lim_{r\to 0} \limsup_{k\to\infty} {}_r h_k(x) \leqq 0$ if $x \notin C$. For any $r > 0$, there is $z \in \text{int } C \cap B_r(x)$. By definition, ${}_r h_k(x) \leqq h_k(z)$ for all k, thus

$$\limsup_{k\to\infty} {}_r h_k(x) \leqq \lim_{k\to\infty} h_k(z) = 0 .$$

Consequently $\lim_{k\to\infty} \limsup_{k\to\infty} {}_r h_k(x) \leqq 0$. This completes the proof. □

Combining Theorems 2.1 and 2.2, we immediately obtain the following theorem. The proof is omitted.

<u>Theorem 2.3.</u> Assume that int C is nonempty. Under conditions

B1 and B2', every sequence $\{ x_k \in M_k \ ; \ k=1,2,\ldots \}$ has a convergent subsequence and every cluster point of the sequence is in M . Moreover, $\lim_{k\to\infty} U_k(x_k) = \min_C f$.

2.5 Examples of Penalty Functions

In this section, we consider the following problem:

(P') minimize $f(x)$ subject to $g_i(x) \leqq 0$, $i=1,\dots,m$,

where $f, g_1, \dots, g_m$ are everywhere finite convex functions on R^n . Let $C = \{ x \in R^n \, ; \, g_i(x) \leqq 0 , \, i=1,\dots,m \}$ and $C_0 = \{ x \in R^n \, ; \, g_i(x) < 0 , \, i=1,\dots,m \}$. It is well known that if C_0 is nonempty, then $C_0 = \text{int } C$. It is now possible to exemplify several classes of penalty functions concerning the infimal convergence condition B2 or B2' for problem (P'). In the following, let $\{ t_k \}$ be a sequence of positive numbers strictly decreasing to zero.

(a) Interior Penalty Functions [F8] [L5]

The *interior penalty functions* are frequently called the *barrier functions* because of their barrier properties to prevent violation of the constraints. It is noted that these functions are defined so long as C_0 is nonempty. Two well-known interior penalty functions are defined as

$$h_k(x) = -t_k \sum_{i=1}^m 1/g_i(x) \quad \text{if } x \in C_0 , \quad = +\infty \quad \text{otherwise}$$

and

$$h_k(x) = -t_k \sum_{i=1}^m \log[-g_i(x)] \quad \text{if } x \in C_0 , \quad = +\infty \quad \text{otherwise}$$

which are called the *inverse barrier function* and the *logarithmic barrier function*, respectively. For both of them, it is easy to verify that $h_k \longrightarrow h$, where $h(x) = 0$ if $x \in C_0$, $= +\infty$ otherwise, and $h_k \xrightarrow[\text{inf}]{} \delta_C$.

(b) Exterior Penalty Functions [F8] [L5]

In *exterior penalty function* methods, no penalty is assigned to feasible points, whereas the penalty for constraint violation increases as parameters change. These functions are sometimes called *loss functions*. One of the most familiar is the *quadratic loss function* defined by

$$h_k(x) = \sum_{i=1}^{m} \{ \max[0, g_i(x)] \}^2 / t_k .$$

Then $h_k \longrightarrow \delta_C$ and $h_k \xrightarrow[\inf]{} \delta_C$.

(c) Exponential Penalty Functions [A4] [E1] [M18]

Another interesting class of penalty functions is the *exponential penalty function* first introduced by Allran and Johnsen [A4]. A general class of these functions is the following [M18]:

$$h_k(x) = \sum_{i=1}^{m} s_k \exp[g_i(x)/t_k]$$

where $0 < t_k \leqq s_k \leqq 1$. Suppose that C_0 is nonempty, then

$$\lim_{k\to\infty} h_k(x) = 0 \quad \text{if} \quad x \in C_0 \ , \quad = +\infty \quad \text{if} \quad x \notin C$$

while $h_k \xrightarrow[\inf]{} \delta_C$.

<u>Remark 2.1</u>. It is noted that a class of multiplier functions defined by Evans and Gould [E2] is a generalized class of exterior and exponential penalty functions. Concerning convex programming problems, Evans-Gould multiplier functions seem almost equivalent in principle to our penalty functions. However, our approach is more general than those because we do not treat

the constraint set in terms of constraint functions.

Remark 2.2. In exterior penalty function methods, it is not required that int C is nonempty. In particular, if each g_i is affine and $C = \{ x \in R^n ; g_i(x) = 0 , i=1,\dots,m \}$, then, for example, the quadratic loss function takes the form

$$h_k(x) = \sum_{i=1}^{m} \{g_i(x)\}^2 / t_k ,$$

moreover, $h_k \longrightarrow \delta_C$ and $h_k \xrightarrow[\inf]{} \delta_C$.

2.6 Conclusion

We have seen that the convergence of various kinds of penalty function methods independently proposed by many authors can be described in a unified manner in terms of the notion of infimal convergence of convex functions.

From a numerical point of view, however, it should be noted that these methods have an unfavorable property that they inevitably become ill-conditioned as the sequential unconstrained minimization proceeds. An attempt to overcome such difficluty will be proposed in the next chapter.

CHAPTER 3

APPLICATION OF FENCHEL DUALITY TO PENALTY METHODS

FOR GENERAL CONVEX PROGRAMS

In this chapter, a new class of sequential unconstrained optimization methods, called the conjugate penalty methods, is proposed for solving convex programming problems. The methods utilize conjugate convex functions and are based on Fenchel's duality theorem in convex analysis. It is shown that, under certain conditions, conjugate penalty functions behave quite mildly and hence avoid the ill-conditioning of ordinary penalty methods of Chapter 2. Convergence of the methods is proved and the relationship between ordinary and conjugate penalty methods is shown.

3.1 Introduction

Recently, many authors have reported a various methods for solving NLP problems by transforming each constrained optimization problem into unconstrained optimization problems [B9] [F8] [L5] [R14]. A characteristic underlying those methods is that a solution of the original problem can be obtained as a limit of sequential solutions to the transformed unconstrained problems. Among those methods, the sequential unconstrained minimization techniques [F8], commonly abbreviated SUMT, have been widely used in practice. They, sometimes called penalty function methods or penalty methods, reduce the computational process to unconstrained minimization of a transformed function, called a penalty function, combining the objective function, the constraint functions and one or more parameters. A foundation of those methods for convex programs is stated in detail in the previous chapter of the thesis. Meanwhile, it is well known that the penalty functions inevitably ill-conditioned near the boundary of the constraint region as the iteration proceeds [L5] [L8]. Indeed, such ill-conditioning causes serious computaional difficulties in solving actual problems.

In this chapter, restricting our attention to convex programs, we present a new class of sequential unconstrained optimization methods which we call conjugate penalty methods. Under appropriate assumptions they circumvent the ill-conditioning of ordinary penalty methods. The idea is to dualize ordinary penalty methods by use of Fenchel's duality theorem [R5] which is stated in Appendix. Specifically, the conjugate penalty method involves sequential unconstrained maximizations of conjugate penalty functions which behave quite mildly near the solution as the sequential maximization proceeds. It is shown that maximizing the conjugate penalty function is dual to minimizing the ordinary penalty functions.

The concept of conjugate convex (concave) functions, originated by Fenchel [F2] and applied to NLP variously, e.g., [B11] [F1] [K5] [R4] [R5], plays a central role in this chapter. The material from convex analysis used in this chapter can be found in Appendix.

This chapter is organized as follows: Section 3.2 shows a duality between two extremum problems derived from a general convex program. Section 3.3 defines the conjugate penalty method and prove its convergence. In Section 3.4, several conditions are given, under which the conjugate penalty functions are well-behaved. Section 3.5 discusses the conjugate interior penalty methods restricted to convex programs with inequality constraints in order to go into further details. Finally, Section 3.6 gives illustrative numerical examples.

3.2 A General Convex Program and Fenchel Duality

Consider the following general convex programming problem:

(P) minimize $f(x)$ subject to $x \in C$,

where f is a closed convex function on R^n and C is a non-empty closed convex set in R^n . The convex programming problem (P) is equivalent to the following 'unconstrained' problem:

$$\text{minimize} \quad U(x) \triangleq f(x) - \gamma_C(x) \quad \text{over} \quad x \in R^n , \tag{3.1}$$

where γ_C is the indicator function of C defined by

$$\gamma_C(x) = 0 \quad \text{if} \quad x \in C , \quad = -\infty \quad \text{if} \quad x \notin C .^{\dagger}$$

Obviously, γ_C is a closed concave function on R^n .

We assume the following:

<u>C1</u>. The finite minimum of U is uniquely attained at $\bar{x}$. Namely $\bar{x}$ is the unique minimum of problem (P).

<u>C2</u>. $\mathrm{ri}(\mathrm{dom}\ f)$ and $\mathrm{ri}\ C$ have a point in common.

In order to guarantee the existence of a minimum (possibly not unique), we may suppose that f and C have no direction of recession in common [R5 , Thm.27.3]. Some other conditions for the existence of minima are found in [A3]. The latter assumption is automatically satisfied when f is finite everywhere, i.e., $\mathrm{dom}\ f = R^n$, and $\mathrm{ri}\ C$ is nonempty, as is almost the case in practical problems.

Let V be a closed concave function defined by

$$V(y) \triangleq \gamma_C^*(y) - f^*(y) , \tag{3.2}$$

where f^* and γ_C^* are conjugates of f and γ_C , respectively.

$^{\dagger}\gamma_C$ is the negative of the indicator function δ_C of Chapter 2.

The function γ_C^* is the negative of the support function of C [R5 , p.28] and, hence, a positively homogeneous closed concave function.

The following lemma is derived from Fenchel's duality theorem.

Lemma 3.1. Let U and V be defined by (3.1) and (3.2), respectively. If assumptions C1 and C2 are satisfied, then

$$f(\bar{x}) = \min_{x \in R^n} U(x) = \sup_{y \in R^n} V(y) ,$$

where the supremum is attained. Moreover, the maximum set of V is $\partial f(\bar{x}) \cap \partial\gamma_C(\bar{x})$, and conversely, for each maximum $\bar{y}$ of V , the set $\partial f^*(\bar{y}) \cap \partial\gamma_C^*(\bar{y})$ is a singleton $\{\bar{x}\}$, where ∂ denotes the subdifferential operator.

Proof. The first part is immediate from Fenchel's duality theorem with $g(x) = \gamma_C(x)$. Now prove the latter half. We can show that the following conditions are equivalent;

(i) $f(\bar{x}) - \gamma_C(\bar{x}) = \gamma_C^*(\bar{y}) - f^*(\bar{y})$;

(ii) $f(\bar{x}) + f^*(\bar{y}) = \langle\bar{x},\bar{y}\rangle = \gamma_C(\bar{x}) + \gamma_C^*(\bar{y})$;

(iii) $\bar{y} \in \partial f(\bar{x}) \cap \partial\gamma_C(\bar{x})$;

(iv) $\bar{x} \in \partial f^*(\bar{y}) \cap \partial\gamma_C^*(\bar{y})$. ([R5 , Thm.23.5])

Let x' be an arbitrary point in $\partial f^*(\bar{y}) \cap \partial\gamma_C^*(\bar{y})$, then obviously $f(x') - \gamma_C(x') = f(\bar{x}) - \gamma_C(\bar{x})$. Thus, $\partial f^*(\bar{y}) \cap \partial\gamma_C^*(\bar{y})$ must be a singleton $\{\bar{x}\}$ by the uniqueness. □

The minimum set $\partial f(\bar{x}) \cap \partial\gamma_C(\bar{x})$ of V is clearly a closed convex set. In particular, when f is differentiable, the maximum set of V is fairly simplified by the following.

Lemma 3.2. Let all requirements in Lemma 3.1 be satisfied. In addition, if f is differentiable at $\bar{x}$, then the supremum of V is uniquely attained at $\bar{y} = \nabla f(\bar{x})$.

Proof. Since $\partial f(\bar{x}) = \{ \nabla f(\bar{x}) \}$ and the maximum set of V is nonempty by Lemma 3.1, it is necessarily $\{ \nabla f(\bar{x}) \}$. $\square$

3.3 Conjugate Penalty Methods

There are a number of methods that solve problem (P) by transforming it into a sequence of unconstrained problems of the form:

(P_k) minimize $U_k(x) \triangleq f(x) - h_k(x)$ over $x \in R^n$.

For each $k = 1, 2, \ldots$, each auxiliary problem (P_k) is solved and the optimal solution to problem (P) is obtained as a limit of a sequence of the optimal solutions to problem (P_k). According to the types of h_k , functions U_k are classified into several classes, e.g., barrier functions, loss functions, and exponential penalty functions [F8][L5][L8][R14]. Here, we call those functions generically *penalty functions*. The penalty functions U_k should be constructed so that:

A1. For every k , there exists a (unique) x_k that minimizes U_k over R^n .

A2. x_k converges to $\bar{x}$ as $k \to \infty$, and the limit of $U_k(x_k)$ is the minimum value of problem (P).

In the following, let appropriate conditions be implicitly assumed so that the properties A1 and A2 above are fulfilled. Such conditions are studied in Chapter 2 of this thesis. Moreover, we assume the following:

C3. Each h_k is a closed concave function with ri(dom f) $\cap$ ri(dom h_k) $\neq \emptyset$;

C4. int(dom f) $\neq \emptyset$ and f is differentiable on int(dom f) ;

C5. Every x_k , a minimum of U_k , belongs to int(dom f) .

Let us define the *conjugate penalty functions* V_k on R^n by

$$V_k(y) \triangleq h_k^*(y) - f^*(y) , \tag{3.3}$$

where f^* and h_k^* are the conjugates of f and h_k , respectively. Now consider a sequence of problems:

(Q_k) maximize $V_k(y)$ over $y \in R^n$.

Since f^* and h_k^* are convex and concave respectively, each problem (Q_k) is to find an unconstrained maximum of the concave function V_k .

It is now possible to define a conjugate penalty method for solving problem (P) in a manner quite similar to that in ordinary penalty methods. Specifically, we try to solve problem (P) by successive maximizations of the conjugate penalty functions V_k , $k = 1, 2,\ldots$. Therefore, the methods may be regarded as one of the SUMT procedures.

The following theorem proves convergence of the conjugate penalty method.

Theorem 3.1. Let V_k be defined by (3.3). If assumptions C1 through C5 are satisfied, then there exists for every k a unique maximum y_k of V_k and $y_k = \nabla f(x_k)$, where x_k is a minimum of problem (P_k). Moreover the y_k and $V_k(y_k)$ converge to $\nabla f(\bar{x})$ and $f(\bar{x})$, respectively, as $k \to \infty$.

Proof. By C3, the existence of a maximum y_k of V_k follows from Fenchel's duality theorem. By the differentiability of f , it follows from an argument similar to Lemma 3.2 that the y_k is unique and is equal to $\nabla f(x_k)$. As x_k converges to $\bar{x}$ and the mapping ∇f is continuous on $\operatorname{int}(\operatorname{dom} f)$, y_k also converges to $\nabla f(\bar{x})$. Since $U_k(x_k)$ converges to $f(\bar{x})$, the convergence of $V_k(y_k)$ to $f(\bar{x})$ follows immediately from the relation

$$U_k(x_k) = \min_x U_k(x) = \max_y V_k(y) = V_k(y_k) . \qquad \square$$

It may be remarked that the conjugate penalty functions are really defined on the dual space of R^n , which is identified with R^n . Thus for optimization problems in more general spaces, it may be possible to consider conjugate penalty functions on the dual spaces.

3.4 Advantage of the Conjugate Penalty Methods

Difficulties in computing minima of ordinary penalty functions result mainly from the fact that the penalty functions U_k grows extremely steep-valleyed near the minimum of the problem as k increases [R14]. Since U_k should converge in a certain sense to the function U, the reason for such irregularity may be that the minimum $\bar{x}$ generally lies on the boundary of dom U.

As regards the function V, however, the maxima of V may be in int(dom V) even when the minimum of U is on the boundary of dom U. In fact, this is true for a certain class of problems. In those problems, the conjugate penalty functions V_k are expected to be uniformly bounded on a neighborhood of the maxima of V. Therefore, we may bypass the difficulty inherent to ordinary penalty methods by employing the conjugate penalty functions to solve problem (P).

In this section, we study conditions on problem (P) for the maxima of V interior to dom V. The necessary and sufficient condition for $\bar{y} \in$ int (dom V) is stated in the following

Theorem 3.2. Assume that assumptions C1 and C2 are satisfied and that f is differentiable at $\bar{x}$. Then, $\bar{y} \in$ int(dom V), if and only if the following two conditions are simultaneously satisfied:

(a) $(f0^+)(x) > \langle \nabla f(\bar{x}), x \rangle$ for every $x \neq 0$;

(b) $\langle \nabla f(\bar{x}), x \rangle > 0$ for every $x \in 0^+C$ and $x \neq 0$;

where $\bar{x}$ is the minimum of U and $\bar{y}$ is the maximum of V.

Proof. First, note that $\bar{y} \in \operatorname{int}(\operatorname{dom} V)$ if and only if $\bar{y} \in \operatorname{int}(\operatorname{dom} f^*)$ and $\bar{y} \in \operatorname{int}(\operatorname{dom} \gamma_C^*)$ simultaneously. It follows from [R5 , Cor.13.3.4(c)] that $\bar{y} \in \operatorname{int}(\operatorname{dom} \gamma_C^*)$ if and only if $(f0^+)(x) - \langle \bar{y},x \rangle > 0$ for every $x \neq 0$. This is exactly the condition (a) since $\bar{y} = \nabla f(\bar{x})$ by Lemma 3.2. On the other hand, taking account of the concavity of γ_C , the necessary and sufficient condition for $\bar{y} \in \operatorname{int}(\operatorname{dom} \gamma_C^*)$ is

$$(\gamma_C 0^+)(x) - \langle \bar{y},x \rangle < 0 \quad \text{for every} \quad x \neq 0 ,$$

which reduces to

$$\langle \bar{y},x \rangle > 0 \quad \text{for every nonzero} \quad x \in 0^+C ,$$

because $(\gamma_C 0^+)(x) = 0$ when $x \in 0^+C$, $= -\infty$ otherwise. □

Condition (a) in Theorem 3.2 can be geometrically interpreted as follows: The hyperplane $z = \langle \nabla f(\bar{x}), x-\bar{x} \rangle + f(\bar{x})$ in R^{n+1} supports epi f at $\bar{x}$ but the set of points at which the hyperplane contacts with epi f is bounded. On the other hand, condition (b) says that either C is compact, i.e. $0^+C = \{0\}$, or there is no halfline orthogonal to $\nabla f(\bar{x})$ which emanates from $\bar{x}$ and is contained in C .

In general, the negative of the polar of the convex cone generated by dom γ_C^* is the recession cone of γ_C . Dually, the negative of the polar of the recession cone of γ_C is the closure of the convex cone generated by dom γ_C^* [R5 , Thm.14.2]. While the recession cone of γ_C is 0^+C and the convex cone generated by dom γ_C^* is dom γ_C^* itself, because γ_C^* is a positively homogeneous closed concave function. Consequently, dom γ_C^* is the negative of the polar of the cone 0^+C .

In particular, if V is finite everywhere, i.e., $\text{dom } V = R^n$, then of course $\bar{y} \in \text{int}(\text{dom } V)$ holds. The following theorem states a necessary and sufficient condition for V everywhere finite.

Theorem 3.3. V is finite everywhere if and only if the following conditions are simultaneously satisfied:

(a) f is co-finite, i.e., $(f0^+)(x) = +\infty$ for every $x \neq 0$;

(b) C is compact, i.e., $0^+C = \{0\}$.

Proof. Note that V is finite everywhere if and only if both f^* and γ_C^* are finite everywhere. From [R5 , Cors.13.3.1 and 13.3.2], the theorem follows immediately. □

The co-finiteness of f implies that $\text{epi } f$ contains no non-vertical halflines. This condition is satisfied, of course, if $\text{dom } f$ is compact.

We have considered so far the convex programs with general constraint sets. Now let us consider the problem

$$(\text{P}') \qquad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad g_i(x) \geqq 0 \,, \quad i=1,\ldots,m \,,$$

where f and $-g_i$, $i=1,\ldots,m$, are closed convex functions on R^n. The convex programming problem (P') frequently encountered in practice is a typical case of the (abstract) program (P), in which the constraint set C is specified by the system of inequalities $C = \{x \in R^n ;\ g_i(x) \geqq 0 \,,\ i=1,\ldots,m\}$. Note that C may be rewritten as $C = \{x \in R^n ;\ g(x) \geqq 0\}$, where g is a closed concave function defined by $g(x) = \inf_{1 \leqq i \leqq m} g_i(x)$.

In order to state the results obtained in the earlier part of this section, we should represent the function γ_C^* in terms of the constraint function g_i. By virtue of [R5, Thms.13.5 and 16.5], γ_C^* is the closure of the positively homogeneous concave function generated by g^* or $\mathrm{cl}(\mathrm{conv}_{1 \leqq i \leqq m}\, g_i^*)$, where cl of a function is a function whose epigraph is the closure of the function, and conv of functions is a function whose epigraph is the convex hull of the functions. Namely,

$$\gamma_C^*(y) = \mathrm{cl}\,\{ \sup_{\lambda \geqq 0} g^*\lambda \}(y)$$

$$= \mathrm{cl}\,\{ \sup_{\lambda \geqq 0} \mathrm{cl}(\mathrm{conv}_{1 \leqq i \leqq m}\, g_i^*\lambda)\}(y) \,. \; \dagger$$

However, such expressions are somewhat complicated and impractical.

In the following, we derive a simple sufficient condition on f and g_i that assures condition (b) in Theorem 3.2. Then such a condition, if it exists, together with condition (a) in Theorem 3.2 will imply $\bar{y} \in \mathrm{int}(\mathrm{dom}\, V)$.

For simplicity, we make the following assumption on (P'):

<u>C6</u>. $f, g_1, \ldots, g_m$, are differentiable at $\bar{x}$, the unique minimum of (P').

Then it is easy to see that $0^+C = \bigcap_{i=1}^m 0^+C_i$, since $C = \bigcap_{i=1}^m C_i$, where $c_i = \{ x \,;\, g_i(x) \geqq 0 \}$. In general, for any concave function g, the direction of recession z satisfies the inequality

$$\langle \nabla g(x), z \rangle \geqq 0$$

---

† It should be noted that this expression is valid even for cases in which the number of constraints is not necessarily finite.

for every x at which g is differentiable [R1 , p.383]. Therefore, since each constraint function g_i is differentiable at $\bar{x}$, $x \in 0^+C = \bigcap_{i=1}^{m} 0^+C_i$ implies

$$\langle \nabla g_i(\bar{x}), x \rangle \geqq 0 , \quad i=1,\dots,m .$$

In other words, 0^+C is contained by the negative of the polar of the cone generated by $\{ \nabla g_1(\bar{x}), \dots, \nabla g_m(\bar{x}) \}$. From this, we have a sufficient condition that guarantees condition (b) in Theorem 3.2.

<u>Theorem 3.4</u>. Suppose assumption C6 is satisfied. If every non-zero vector x such that

$$\langle \nabla g_i(\bar{x}), x \rangle \geqq 0 , \quad i=1,\dots,m ,$$

satisfies the inequality

$$\langle \nabla f(\bar{x}), x \rangle > 0 ,$$

then the same inequality holds for any non-zero vector $x \in 0^+C$.

<u>Proof</u>. Immediate from

$$0^+C \subset \{ x ; \langle \nabla g_i(\bar{x}), x \rangle \geqq 0 , i=1,\dots,m \} . \qquad \square$$

It may be noted that the condition in Theorem 3.4 is fairly strong. In fact, there are problems in which some vector x, such that $\langle \nabla g_i(\bar{x}), x \rangle \geqq 0$ for *all* i, does not satisfy $\langle \nabla f(\bar{x}), x \rangle > 0$, while condition (b) in Theorem 3.2 is satisfied. For instance, consider the problem

$$\begin{array}{ll} \text{minimize} & (x_1 + 1)^2 + x_2^2 \\ \text{subject to} & x_1 - x_2^2 \geqq 0 \\ \text{and} & 1 - x_1 - x_2 \geqq 0 . \end{array}$$

Obviously, the solution is $\bar{x} = (\bar{x}_1, \bar{x}_2) = (0,0)$ and $\nabla f(\bar{x}) = (2,0)$, $\nabla g_1(\bar{x}) = (1,0)$, $\nabla g_2(\bar{x}) = (0,-1)$. Taking $x = (0,-1)$, we see that the condition in Theorem 3.4 is not met. However, this problem satisfies condition (b) in Theorem 3.2, because the constraint set is compact.

3.5 A Conjugate Interior Penalty Method

In this section, we consider again the following convex programming problem:

$$(\mathrm{P}') \qquad \text{minimize} \qquad f(x)$$
$$\qquad \text{subject to} \qquad g_i(x) \geqq 0 \ , \quad i=1,\ldots,m \ ,$$

where f and $-g_i$, $i=1,\ldots,m$, are assumed to be everywhere finite convex functions on R^n .

Let

$$C = \{ x \in R^n \ ; \ g_i(x) \geqq 0 \ , \ i=1,\ldots,m \} \ ,$$
$$C_0 = \{ x \in R^n \ ; \ g_i(x) > 0 \ , \ i=1,\ldots,m \} \ .$$

Since the g_i's are everywhere finite concave functions, C is closed and convex and C_0 is open and convex. Furthermore, if C_0 is nonempty, then $C_0 = \text{int } C$ and $C = \text{cl } C_0$.

It is assumed that the following conditions are satisfied in problem (P'):

C7. f is co-finite, i.e., the epigraph of f contains no nonvertical halflines;

C8. C is compact and C_0 is nonempty.

Define the class I_m of extended-real-valued functions as follows: $G \in I_m$ if

(i) G is a continuous concave function with $\text{dom } G = R^m_+$, where R^m_+ is the (strictly) positive orthant in R^m ;

(ii) G is nondecreasing, i.e., for $\xi, \eta \in R^m$, $\xi \leqq \eta$ implies $G(\xi) \leqq G(\eta)$.

Note that (i) implies that $G(\xi)$ tends to $-\infty$ if ξ approaches the boundary of R^m_+ .

An interior penalty method for solving problem (P') is defined for each $G \in I_m$. In what follows, let $G \in I_m$ be given. Define a function h on R^n by

$$h(x) \triangleq G(g_1(x),\ldots,g_m(x)) .$$

Then by the concavity of g_i together with the properties of G, h is a concave function with $\mathrm{dom}\ h = C_0$. Furthermore, $h(x)$ tends to $-\infty$ as x approaches the boundary of C_0.

Let

$$u_t \triangleq \inf_{x\in R^n} U_t(x) \tag{3.4}$$

and

$$S_t \triangleq \{x ; U_t(x) = u_t\}$$

for all $t \geqq 0$, where

$$U_t \triangleq \begin{cases} f - \gamma_C & \text{if } t = 0 , \\ f - th & \text{if } t > 0 , \end{cases}$$

where γ_C is the concave indicator function as defined in Section 3.2. Clearly, the U_t are convex functions with $\mathrm{dom}\ U_t = C$ if $t = 0$, $\mathrm{dom}\ U_t = C_0$ if $t > 0$, and the S_t are convex subsets of $\mathrm{dom}\ U_t$. Minimizing U_0 over R^n is equivalent to solving problem (P'). The functions U_t with a parameter $t > 0$ are ordinary *interior* penalty functions for problem (P'). From the arguments developed in the previous chapter, it is easily seen that the functions U_t infimally converge to U_0 as t decreases to zero. Moreover, as is well known, u_t and S_t converge to u_0 and S_0, respectively, as t decreases to zero [F8][G10][L5][L8].

Now we introduce a family of problems dual to (3.4) by means of

$$v_t \triangleq \sup_{y \in R^n} V_t(y) \tag{3.5}$$

and

$$T_t \triangleq \{ y \; ; \; V_t(y) = v_t \}$$

for all $t \geq 0$, where

$$V_t \triangleq \begin{cases} \gamma_C^* - f^* & \text{if } t = 0 \, , \\ h^*t - f^* & \text{if } t > 0 \, , \end{cases}$$

where the right scalar multiplication of h^* is defined as

$$(h^*t)(y) \triangleq th^*(t^{-1}y) \, , \qquad t > 0 \, .$$

Note that $h^*t = (th)^*$ [R5 , Thm.16.1]. The functions V_t , $t > 0$, are called the *conjugate interior penalty functions* for problem (P').

Remark 3.1. We consider here a continuous version of penalty methods. Hence we adopt the notation for subscripts of U and V in a manner somewhat different from that in Section 5.3. This should, however, cause no difficulty, since the distinction is clear from the context.

Lemma 3.3. Under conditions C7 and C8, the V_t are everywhere finite and concave for all $t \geq 0$. Furthermore,

$$V_0(y) = \lim_{t \downarrow 0} V_t(y) \qquad \text{for every } y \in R^n \, .$$

Proof. By C7, f^* is everywhere finite and convex [R5 , Cor.13.3.1]. Since $\text{dom}\, \gamma_C = C$ and $\text{dom}\, th = C_0$ which are nonempty and bounded by C8, γ_C^* and h^*t are everywhere finite and concave [R5 , Cor. 13.3.1]. Hence, the first part of the lemma follows. The latter half follows from the fact that

$$\gamma_C^*(y) = \gamma_{C_0}^*(y) = \gamma_{\text{dom}\, h}^*(y)$$

$$= (h^*0^+)(y) \qquad [R5, \text{Thm.}13.3]$$

$$= \lim_{t\downarrow 0} (h^*t)(y) \qquad [R5, \text{Cor.}8.5.2]$$

for every $y \in R^n$. This completes the proof. □

This lemma says that as t decreases to zero the conjugate penalty functions $\{V_t\}$ converge pointwise to V_0 which is finite everywhere.

The following theorem demonstrates the relationship between the minima of problem (3.4) and maxima of problem (3.5).

Theorem 3.5. Assume that conditions C7 and C8 are satisfied. Then S_t and T_t are nonempty and compact, and

$$-\infty < u_t = v_t < +\infty$$

for every $t \geqq 0$. Furthermore, the following (i), (ii), (iii) are equivalent for each $t > 0$, and (i), (ii'), (iii') are equivalent for $t = 0$:

(i) $x \in S_t$ and $y \in T_t$;

(ii) $x \in \partial(h^*t)(y) \cap \partial f^*(y)$; (ii') $x \in \partial\gamma_C^*(y) \cap \partial f^*(y)$;

(iii) $y \in \partial f(x) \cap \partial(th)(x)$; (iii') $y \in \partial f(x) \cap \partial\gamma_C(x)$.

Proof. Since f is everywhere finite, f^* is co-finite [R5, Cor.13.3.1]. From this and C7, both U_t and V_t are co-finite and, in particular, have no directions of recession for all $t \geqq 0$. Hence, S_t and T_t are nonempty and compact [R5, Thm.27.1]. The assertion about optimal values follows from Fenchel's duality theorem, for by C8 and Lemma 3.3, conditions (a) and (b) in [R5, Thm.31.1] are satisfied. Therefore,

$$-\infty < u_t = \min U_t = \max V_t = v_t < +\infty$$

for all $t \geqq 0$. Finally, we shall prove the equivalence for $t > 0$. A necessary and sufficient condition for (i) to hold is

$$y \in \partial f(x) \quad \text{and} \quad x \in \partial(h^*t)(y) \qquad [R5 , p.333] .$$

This is equivalent to (ii) and to (iii) by [R5 , Thm.23.5]. The equivalences for $t = 0$ follow analogously. This completes the proof. □

By virtue of Theorem 3.5, for each t the minimum of U_t can be obtained in terms of the maximum of V_t , and vice versa. Therefore, the two sequences of minimization problems (3.4) and maximization problems (3.5) are essentially equivalent, since they are convertible to each other without loss of equality.

The following theorem describes a convergence property enjoyed by maxima of the conjugate interior penalty functions.

Theorem 3.6. Assume that conditions C7 and C8 are satisfied. Then

$$v_0 = \lim_{t \downarrow 0} v_t$$

and

$$0 = \lim_{t \downarrow 0} \sup_{z \in T_t} \inf_{y \in T_0} \| z - y \| .$$

Proof. We shall prove the last equality; then the first equality follows from Lemma 3.3 and the continuity of V_0 . Let us assume that there exist a decreasing null sequence $\{t_k\}$ of positive numbers and an $\varepsilon > 0$ such that

$$\exists\ z_k \in T_{t_k} \quad \text{and} \quad \inf_{y \in T_0} \| z_k - y \| > \varepsilon \qquad \text{for all } k .$$

Let T^{ε} be the boundary of the set $T_0 + \varepsilon B$, where B is the unit sphere in R^n . Since T_0 is compact, T^{ε} is also compact. Choosing $\bar{y}$ arbitrarily in T_0 , let $\tilde{z}_k$ be a point where the line segment joining $\bar{y}$ and z_k intersects T^{ε} . Then $\{\tilde{z}_k\}$ has a convertent subsequence by the compactness of T^{ε} . We assume without loss of generality that $\tilde{z}_k$ converges to $\tilde{z}$. For all k , by the definition

$$V_{t_k}(z_k) \geqq V_{t_k}(\bar{y})$$

from which we have

$$V_{t_k}(\tilde{z}_k) \geqq V_{t_k}(\bar{y})$$

by the concavity of V_{t_k} . Taking the limit as $t \to \infty$, we have

$$V_0(\tilde{z}) \geqq V_0(\bar{y}) ,$$

because by Lemma 3.3 and [R5 , Thm.10.7], $V_t(y)$ is jointly continuous in y and t . But $\tilde{z} \notin T_0$, which is a contradiction. □

Theorem 3.6 implies that the point-to-set map T_t is upper semicontinuous (u.s.c.) at $t = 0$ [B6 , p.109]. In particular, if T_0 is a singleton, say $\{\bar{y}\}$, then every sequence $\{y_k \in T_{t_k}\}$ converges to $\bar{y}$. Note that the first part of Theorem 3.6 could also be deduced from the fact that $u_0 = \lim_{t \downarrow 0} u_t$ and $u_t = v_t$ for all $t \geqq 0$.

<u>Remark 3.2.</u> The conjugate penalty functions $\{V_t\}$ converge to V_0 , which is everywhere finite but in general not everywhere differentiable. It can be shown, however, that the V_t are actually everywhere differentiable for all $t > 0$, provided f and

$-h$ are strictly convex on their effective domains $\operatorname{dom} f = R^n$ and $\operatorname{dom} h = \operatorname{int}(\operatorname{dom} h)$ [R5, Thm.26.3]. In such cases, first derivative methods may be used in each unconstrained maximization of V_t. Furthermore, for each $t > 0$ and $y \in T_t$, necessarily $\nabla f^*(y) = \nabla(h^*t)(y)$. Consequently, from (ii) in Theorem 3.5, $x \in S_t$ may be written as $x = \nabla f^*(y) = \nabla(h^*t)(y)$. More generally, if either f^* or h^* is differentiable, (ii) in Theorem 3.5 reduces to either $x = \nabla f^*(y)$ or $x = \nabla(h^*t)(y)$. In such cases the conversion of y into x may be considerably simplified. We note here that $\nabla(h^*t)(y) = \nabla h^*(t^{-1}y)$. When V_t is nondifferentiable, some suitable method for maximizing nondifferentiable functions should be employed (cf. [F13]).

Remark 3.3. In general, it may not be so easy to evaluate the conjugate penalty functions, because the class of functions for which the conjugate has a simple closed form is limited. However, when the functions possess certain structure, the difficulty might be relaxed somewhat by means of various dual operations [R5, §16]. For instance, if h is separable, i.e.,

$$U_t(x) = f(x) - t\sum_{i=1}^{m} G_i(g_i(x))$$

$$= f(x) - t\sum_{i=1}^{m} h_i(x) \qquad \text{for } t > 0,$$

where $G_i \in I_1$, $i=1,\ldots,m$, then V_t may be written as

$$V_t(y) = \left(t\sum_{i=1}^{m} h_i\right)^*(y)$$

$$= t\sup\left\{\sum_{i=1}^{m} h_i^*(y^i) \,\middle|\, \sum_{i=1}^{m} y^i = y/t\right\} - f^*(y)$$

[R5, Thm.16.4].

We should mention that the evaluation of V_t above is sometimes expensive in practical computation, because one needs to solve constrained subproblems. However, the difficulty can be eliminated provided one is willing to increase the dimensionality of variables. In fact, we have

$$\sup_y V_t(y) = \sup_{y^1,\dots,y^m} \left\{ t \sum_{i=1}^m h_i^*(y^i) - f^*\left(t \sum_{i=1}^m y^i \right) \right\} ,$$

where the maximization is completely unconstrained in R^{mn} .

Remark 3.4. For ordinary penalty methods, the convergence rate analysis has been well investigated [L5]. Here the rate of the conjugate penalty method is briefly discussed. We assume that f and f^* are differentiable and that ∇f and ∇f^* are Lipschitz continuous on some neighborhoods of $\bar{x}$ and $\bar{y}$, respectively, where $\{\bar{x}\} = S_0$ and $\{\bar{y}\} = T_0$. Then there exist positive numbers M_1 and M_2 such that for every $t > 0$ sufficiently small

$$\| y_t - \bar{y} \| = \| \nabla f(x_t) - \nabla f(\bar{x}) \| \leqq M_1 \| x_t - \bar{x} \|$$

and

$$\| x_t - \bar{x} \| = \| \nabla f^*(y_t) - \nabla f^*(\bar{y}) \| \leqq M_2 \| y_t - \bar{y} \| ,$$

where $x_t \in S_t$ and $y_t \in T_t$. Hence, we have

$$\frac{1}{M_1} \leqq \frac{\| x_t - \bar{x} \|}{\| y_t - \bar{y} \|} \leqq M_2$$

for all $t > 0$ sufficiently small. Consequently, we may conclude that $\{y_t\}$ converges to $\bar{y}$ as fast as $\{x_t\}$ does. For instance, if U_t is the logarithmic interior penalty function, we have

$$\| y_t - \bar{y} \| = O(t) .$$

3.6 Numerical Examples

Let us consider the problem

$$\text{minimize} \quad f(x) = \frac{1}{2} \langle x, Ax\rangle + \langle b, x\rangle \tag{3.6}$$

$$\text{subject to} \quad g(x) = r - \langle x, Dx\rangle \geqq 0 ,$$

where A and D are positive definite $n \times n$ matrices, b is an n-vector and r is a positive number.

The logarithmic penalty methods [F8] is to solve a sequence of problems of the form

$$\text{minimize} \quad U_k(x) = f(x) - t_k \log g(x) \quad \text{over} \quad x \in R^n, \tag{3.7}$$

where $\{t_k\}$ is a strictly decreasing sequence of positive numbers converging to zero. Put

$$h(x) = \log g(x)$$

and for $k=1,2,\ldots$,

$$h_k(x) = t_k \, h(x) .$$

Corresponding to (3.7), we can define the conjugate penalty function as

$$\begin{aligned} v_k(y) &= h_k^*(y) - f^*(y) \\ &= t_k \, h^*(y/t_k) - f^*(y) . \end{aligned} \tag{3.8}$$

For problem (3.6), by direct calculation, we have

$$f^*(y) = \frac{1}{2} \langle y-b, A^{-1}(y-b)\rangle \tag{3.9}$$

and

$$h^*(y) = 1 - (1 + r\langle y, D^{-1}y\rangle)^{1/2}$$

$$+ \log \frac{1 + (1 + r\langle y, D^{-1}y\rangle)^{1/2}}{2tr} . \tag{3.10}$$

Substituting (3.9) and (3.10) into (3.8), we have

$$V_k(y) = \log [\, t_k + (t_k + r\langle y, D^{-1}y\rangle)^{1/2} \,] - (\, t_k^2 + r\langle y, D^{-1}y\rangle)^{1/2}$$

$$- \frac{1}{2} \langle y-b, A^{-1}(y-b)\rangle + t_k \,(\, 1 - \log 2rt_k \,) \;.$$

In the following, we give two illustrative numerical exaples of this type.

<u>Example 3.1.</u> Consider the following one-dimensional problem:

$$\begin{array}{ll} \text{minimize} & (x-2)^2 \\ \text{subject to} & 1 - x \geqq 0 \,, \\ & 1 + x \geqq 0 \,. \end{array}$$

The logarithmic interior penalty function is of the form

$$\begin{aligned} U_k(x) &= (x-2)^2 - t_k \log (1 - x^2) & \text{if } -1 < x < 1 \,, \\ &= +\infty & \text{otherwise.} \end{aligned}$$

The conjuagte penalty function becomes

$$V_k(y) = t_k - (\, t_k^2 + y^2 \,)^{1/2} + t_k \log \frac{t_k + (\, t_k^2 + y^2 \,)^{1/2}}{2t_k}$$

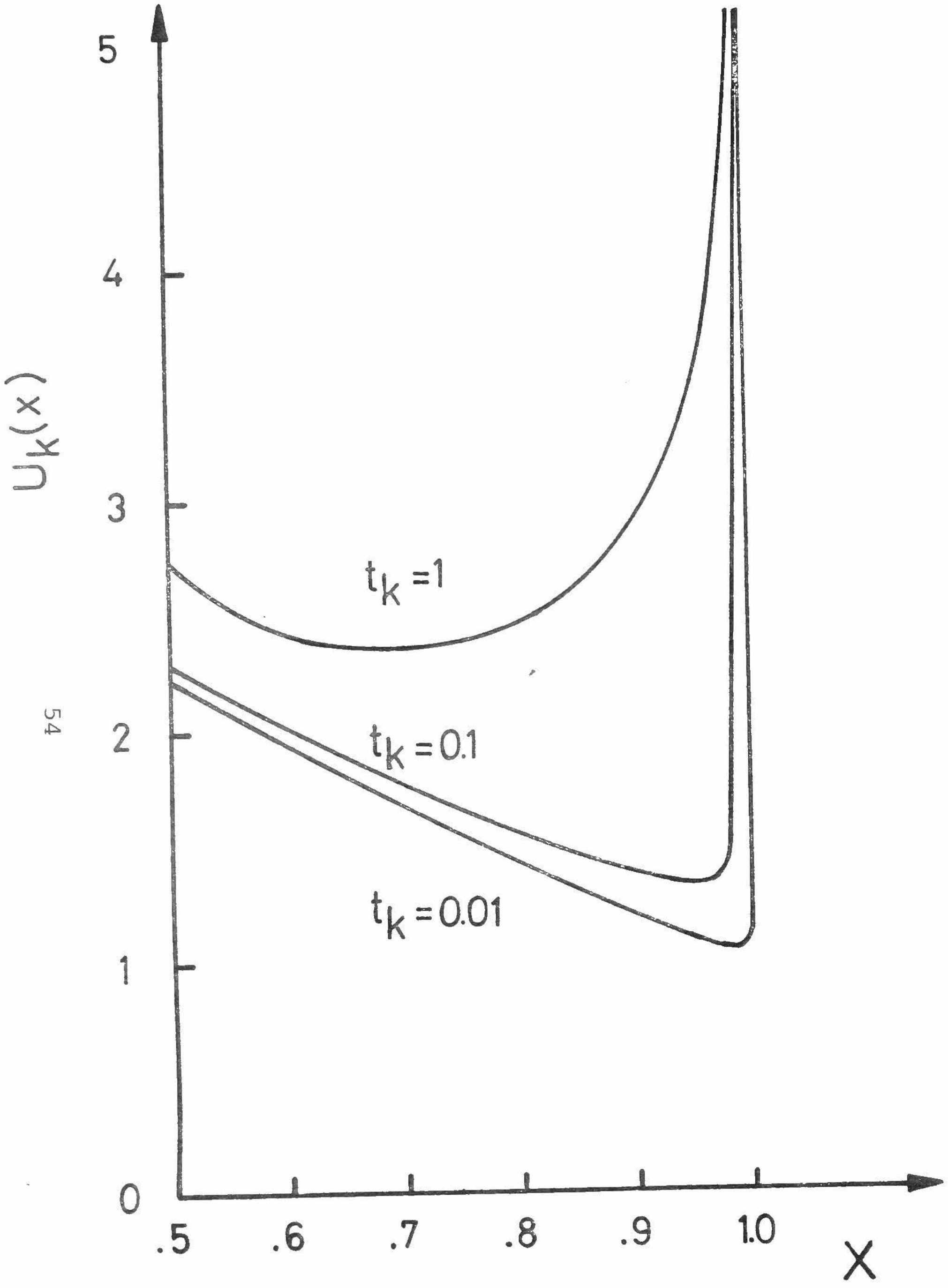
$$- 2y - \frac{y^2}{4}$$

for all $y \in R^n$. Fig. 3.1 and Fig. 3.2 illustrate U_k and V_k, respectively, for some values of t_k. It is seen that V_k is more moderate than U_k for small t_k. Obviously the optimal solution of the problem is $\bar{x} = 1$ and $f(\bar{x}) = 1$.

<u>Example 3.2.</u> Consider the problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x_1^2 + x_2^2 - x_1 x_2 - 7x_1 - 7x_2 \\ \text{subject to} & 25 - 4x_1^2 - x_2^2 \geqq 0 \,. \end{array}$$

As in Example 3.1, we employ the logarithmic interior penalty function. Then, by the calculation, the conjugate penalty func-


5
4
3
2
1
0
$U_k(x)$
$t_k = 1$
$t_k = 0.1$
$t_k = 0.01$
.5
.6
.7
.8
.9
1.0
X

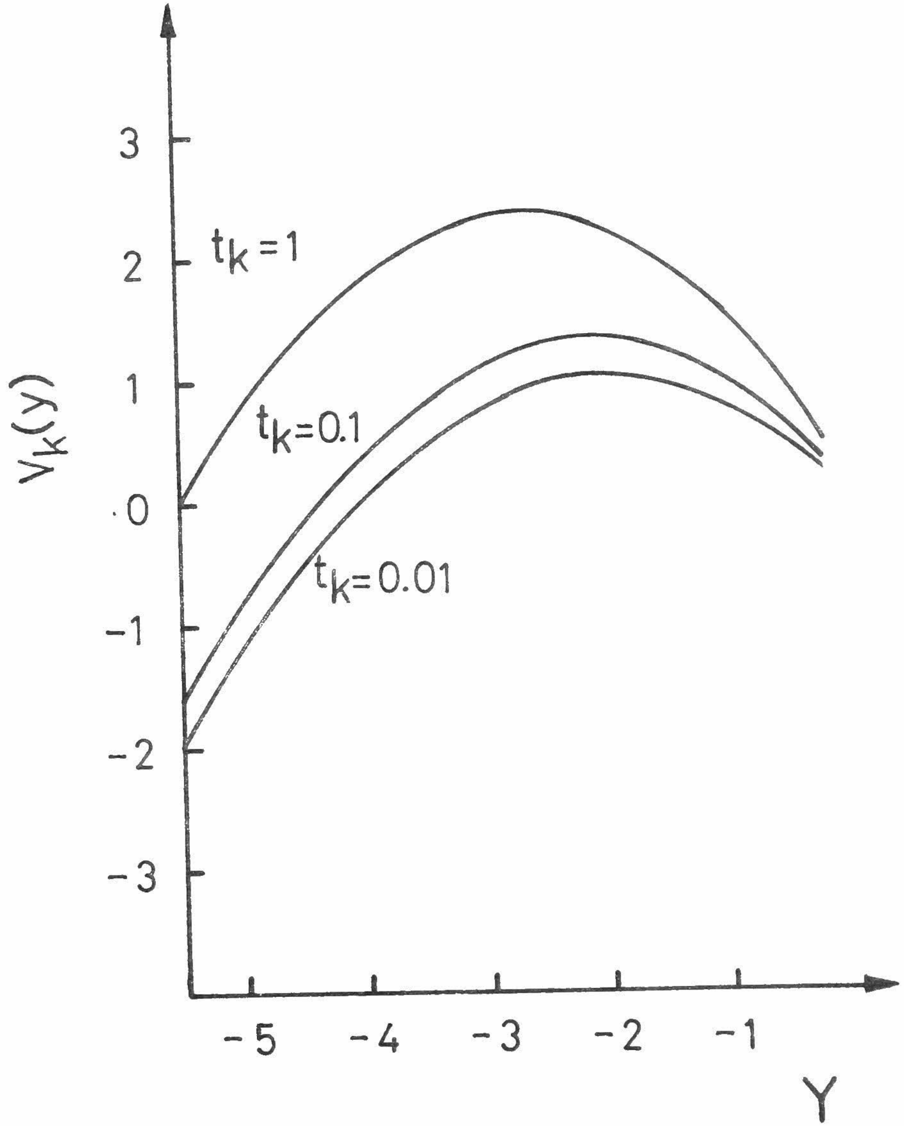

3
2
1
0
−1
−2
−3
$V_k(y)$
$t_k = 1$
$t_k = 0.1$
$t_k = 0.01$
−5
−4
−3
−2
−1
Y

FIG. 3.2

tion is

$$V_k(y) = t_k - p_{t_k}(y) + t_k \log \frac{t_k + p_{t_k}(y)}{50t_k}$$

$$- y_1^2 - \frac{1}{2} y_2^2 - y_1 y_2 - 21y_1 - 14y_2 - 122.5 ,$$

where $y = (y_1, y_2)$ and

$$p_{t_k}(y) = [t_k^2 + 25(y_1^2 / 4 + y_2^2)]^{1/2} .$$

The optimal solution of the problem is $\bar{x} = (2,3)$ and $f(\bar{x}) = -30$.

The computational results for Examples 3.1 and 3.2 are presented in Tables 3.1 and 3.2, respectively. In each example, the initial point is the origin for $t = 1$, and the subsequent unconstrained maximizations of V_k are initiated from the previous terminating points. The termination criterion is

$$\| y^{k+1} - y^k \| < 10^{-5} .$$

The values of x_k are given by $x_k = \nabla f^*(y_k)$.

In both examples, V_k are twice differentiable everywhere for all $t_k > 0$. We have used the *pure* Newton's method for the maximization of each V_k . On the other hand, we would not be able to use the method in each unconstrained minimization of the ordinary interior penalty function, since we would always go outside the feasible region when t is small. In conjugate penalty methods, therefore, we can avoid the considerable effort of determining step sizes to maintain feasibility, as is required by ordinary interior penalty methods.

Finally, in conjugate penalty methods, we need not determine an initial point in the interior of the feasible region, as is required by ordinary interior penalty methods.

Table 3.1

t_k	No. of iterations	y_k	max V_k	$x_k = \nabla f^*(y_k)$
1	5	-2.62222	2.36255	0.688892
10^{-1}	3	-2.09326	1.33503	0.953368
10^{-2}	3	-2.00993	1.05610	0.995037
10^{-3}	2	-2.00100	1.00791	0.999500
10^{-4}	2	-2.00010	1.00102	0.999950
10^{-5}	2	-2.00001	1.00013	0.999995
10^{-6}	1	-2.00000	1.00001	0.999999

initial point $y = 0$.

Table 3.2

t_k	No. of iterations	y_k		max V_k	$x_k = \nabla f^*(y_k)$	
1	13	-8.01275	-3.06639	-29.6688	1.90812	2.92087
10^{-1}	9	-8.00139	-3.00667	-29.7388	1.99055	2.99194
10^{-2}	7	-8.00014	-3.00067	-29.9509	1.99904	2.99919
10^{-3}	5	-8.00002	-3.00007	-29.9928	1.99990	2.99991
10^{-4}	3	-8.00000	-3.00001	-29.9990	1.99998	2.99998
10^{-5}	1	-8.00000	-3.00001	-29.9999	1.99998	2.99998

initial point $y = (0,0)$.

3.7 Conclusion

We have presented a new class of sequential unconstrained optimization techniques for the solution of convex programming problems. The method has an advantage over ordinary penalty function methods in that it will circumvent unfavorable boundary properties of ordinary penalty functions, as far as the conditions in Theorem 3.2 are satisfied. Those conditions are met for some classes of problems that are often encountered in practice. For example, a strictly convex objective function and strictly concave constraint functions will form a problem which satisfies both conditions in Theorem 3.2. The present method has a drawback, however, in its practical implementation, because it is not an easy matter to obtain a terse expression of conjugates for any convex and concave functions. This approach may be particularly attractive for the type of problems such as problem (3.6), in which functions have their conjugates in a simple closed form.

CHAPTER 4

APPLICATION OF FENCHEL DUALITY TO MULTIFACILITY LOCATION PROBLEMS

In this chapter, we consider multifacility location problems under a very general setting and develop a dual approach to solve those problems. The multifacility location problem is a typical one of minimizing a convex function which is not everywhere differentiable. Formulation of the dual problem is essentially based on Fenchel's duality theorem which has also played a crucial role in the previous chapter. It can be shown that, for some important classes of problems, the dual problem may be formulated so as to involve only differentiable functions.

4.1 Introduction

Recently much has been studied on the multifacility location problems which have been practically applied to locating facilities such as plants, warehouses and firestations, locating machines in a factory and communication networks. The problem is that of locating some new facilities optimally in relation to existing facilities. Optimality is achieved when the total transportation cost is minimized. The transportation cost from one facility to another is normally assumed to be proportional to suitably weighted distance between the facilities. By assuming such linearity, various computational methods have been proposed by many authors. Those methods may be refered to a bibliography by Francis and Goldstein [F11]. However, there are some problems in which the transportation cost cannot be assumed a linear function of the distances. Thus we generalize here the problem by assuming the transportation cost as a nondecreasing convex function of distances measured by any ℓ_p norm. We call this the generalized multifacility location problem with arbitrary norms. This generalized problem contains of course the traditional multifacility location problem as a special case. It is noted that the functions involved in the generalized multifacility location problem are all convex but, in general, not everywhere differentiable.

In this chapter, we propose a dual approach to the generalized multifacility location problem. Formulation of the dual problem is essentially based on Fenchel's duality theorem which utilizes the conjugates of convex and concave functions. We show that for a certain, often very important, classes of problems the dualization avoids the nondifferentiability of cost functions, which is inherent in the primal problems. Specifically, if each transportation cost is linear with respect to the distance measured by the rectangular norm, then the dual problem may be formulated as a linear programming problem. Similarly, if some transportation costs are the same as that of above and others quadratic with respect to the distance measured by the Euclidean norm, then the dual becomes a quadratic programming problem.

This chapter is organized as follows: In Section 4.2, the multifacility location problem is defined in a very general form.

In Section 4.3, another problem is defined and the duality between these two problems is verified. In Section 4.4, examples are given in order to demonstrate the validity of such dualization. Computation results are placed in Section 4.5.

4.2 The Multifacility Location Problems

The problem to be considered is as follows: There are m facilities situated at points $a_1, a_2, \ldots, a_m$ on a plane. Suppose that n new facilities should be located and that cost should be imposed on transportation of goods between the new facilities and the existing facilities and between the new facilities themselves. Then it is necessary to locate them simultaneously in such a way that the total transportation cost is minimized. This problem is called the *generalized multifacility location problem* and is formally stated as follows:

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{minimize}} \; & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ji}(\|x_j - a_i\|_{p_1}) \\ & + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} g_{jk}(\|x_j - x_k\|_{p_2}) \; , \end{aligned} \tag{4.1}$$

where $x_1, x_2, \ldots, x_n$ are the locations of the new facilities to be found, f_{ji} and g_{jk} are nondecreasing convex functions on $[0,+\infty)$, $\|\cdot\|_p$ is the ℓ_p norm representing the distance between two facilities and p_1 and p_2 are real numbers such that $1 \leqq p_i \leqq \infty$ $(i=1,2)$.

It is noted that the two common ℓ_p norms used are the Euclidean (ℓ_2) norm and the rectangular (ℓ_1) norm, but other norm might also be considered. Furthermore, different norms may be contained simultaneously in problem (4.1), since it is not unusual that more than one kinds of transportation are used in one problem. For example, in a plant layout problem, some materials are supplied through conveyors or pipes among the machines themselves where the Euclidean norm is used as a measure of distance; while among the machines and storages other materials are supplied by folklift trucks operating in a grid of aisles where the rectangular norm is appropriate. We assume here, however, that at most two different norms, ℓ_{p_1} and ℓ_{p_2}, are used for simplicity of presentation.

As a special case of problem (4.1), the next problem might be considered.

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{minimize}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ji} \|x_j - a_i\|_{p_1} \\ & + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \beta_{jk} \|x_j - x_k\|_{p_2} \quad , \end{aligned} \tag{4.2}$$

where α_{ji} and β_{jk} are nonnegative weighting constants.

A wide variety of approaches have been proposed for problem (4.2), in particular, where $n = 1$ or $p_i = 1$ or 2 . (See for example a bibliography by Francis and Goldstein [F11].) Among others, dual approaches have been studied by Love [L6], Planchart and Hurter [P2] and Juel and Love [J3] in recent years. It seems, however, that the generalized problem (4.1) has little been investigated in literature. In the remainder of this chapter, we shall be concerned with the generalized problem (4.1).

4.3 The Dual Problem

Throughout this chapter, we have in mind the following interpretations:

m — the number of existing facilities;

n — the number of new facilities;

i — indices of existing facilities, $1 \leq i \leq m$;

j, k — indices of new facilities, $1 \leq j,k \leq n$;

a_i — a vector in R^2 , the location of existing facility i ;

x_j — a vector in R^2 , the location of new facility j ;

x — a vector in R^{2n} such that $x = (x_1, \ldots, x_n)^T$;

z_{jk} — a vector in R^2 , the direction from the facility k to the facility j , viz. $z_{jk} = x_j - x_k$;

z — a vector in $R^{n(n-1)}$ such that $z = (z_{12}, z_{13}, \ldots, z_{n-1,n})^T$;

A — an $n(n-1)\times 2n$ matrix which defines the relationship between x and z , viz.,

$$z = Ax = \begin{pmatrix} I & -I & 0 & 0 & \cdots & 0 & 0 \\ I & 0 & -I & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ I & 0 & 0 & 0 & \cdots & 0 & -I \\ 0 & I & -I & 0 & \cdots & 0 & 0 \\ 0 & I & 0 & -I & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & I & -I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} z_{12} \\ z_{13} \\ \cdot \\ \cdot \\ z_{1n} \\ z_{23} \\ z_{24} \\ \cdot \\ \cdot \\ z_{n-1,n} \end{pmatrix} \quad (4.3)$$

where I is the 2×2 identity matrix;

v_j, v_j^i — dual vectors in R^2 corresponding to x_j with $\sum_{i=1}^{m} v_j^i = v_j$;

v, v^i — vectors in R^{2n} suth that $v = (v_1,\ldots,v_n)^T$, $v^i = (v_1^i, \ldots, v_n^i)^T$ and $\sum_{i=1}^{m} v^i = v$;

u_{jk} — a dual vector in R^2 corresponding to z_{jk} ;

u — a vector in $R^{n(n-1)}$ such that $u = (u_{12}, u_{13}, \ldots, u_{n-1,n})^T$;

f_{ji} — a nondecreasing convex function on $[0,+\infty)$, the transportation cost between the new facility j and the existing facility i ;

g_{jk} — a nondecreasing convex function on $[0,+\infty)$, the transportation cost between the new facility j and the new facility k ;

p_i, q_i — real numbers such that $1 \leqq p_i , q_i \leqq +\infty$ and $1/p_i + 1/q_i = 1$, $i = 1, 2$.

In this section, we are concerned with the following pair of problems:

$$\text{(PL)} \qquad \underset{x_1,\ldots,x_n}{\text{minimize}} \ \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ji}(\|x_j - a_i\|_{p_1}) + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} g_{jk}(\|x_j - x_k\|_{p_2})$$

and

$$\text{(DL)} \qquad \underset{u,v^1,\ldots,v^m}{\text{maximize}} \ -\sum_{j=1}^{n-1}\sum_{k=j+1}^{n} g_{jk}^{+}(\|u_{jk}\|_{q_2}) - \sum_{i=1}^{m}\sum_{j=1}^{n} [\, \langle a_i, v_j^i\rangle + f_{ji}^{+}(\|v_j^i\|_{q_1})\,]$$

subject to

$$\sum_{i=1}^{m} v^i = A^T u \, .$$

Problem (PL) is the generalized multifacility location problem as stated in Section 4.2. We shall show that problem (DL) is actually the dual of problem (PL) in the sense of Fenchel. Before proceeding to the main theorem, we need the following lemma.

Lemma 4.1. Let f be a convex function on R^n defined by

$$f(x) = g(\|x\|_p) \quad , \qquad 1 \leqq p \leqq +\infty \ ,$$

where g is a nondecreasing convex function with $\operatorname{dom} g = [0,+\infty)$.
Then

$$f^*(v) = g^+(\|v\|_q) \quad ,$$

where $1 \leqq q \leqq +\infty$ and $1/p + 1/q = 1$.

Proof. By the definition,

$$f^*(v) = \sup_x \{ \langle x,v \rangle - f(x) \}$$

$$= \sup_{\xi \geqq 0} \ \sup_{\|x\|_p = \xi} \{ \langle x,v \rangle - g(\xi) \} ,$$

and by Hölder's inequality,

$$= \sup_{\xi \geqq 0} \{ \xi \|v\|_q - g(\xi) \}$$

$$= g^+(\|v\|_q) \ . \qquad \square$$

It is now possible to prove the main result of this chapter.

Theorem 4.1. Problems (PL) and (DL) are dual to each other in the sense of Fenchel. Moreover, the supremum of problem (PL) and the infimum of problem (DL) are equal and both attained.

Proof. We first define the functions f_i , i=1,...,m, and f on R^{2n} as follows:

$$f_i(x) = \sum_{j=1}^{n} f_{ji}(\|x_j - a_i\|_{p_1}) \quad , \qquad i=1,\dots,m \; , \tag{4.4}$$

$$f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ji}(\|x_j - a_i\|_{p_1}) = \sum_{i=1}^{m} f_i(x) \; . \tag{4.5}$$

Next let us define the function h on $R^{n(n-1)}$ by

$$h(z) = \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} g_{jk}(\|z_{jk}\|_{p_2}) \; . \tag{4.6}$$

Since norms are convex and f_{ji} are nondecreasing convex functions, the composite functions f_i and f are convex on R^{2n} [R5 , Thm.5.1]. Similarly, h is a convex function on $R^{n(n-1)}$. It is noted that by the relation (4.3)

$$h(Ax) = \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} g_{jk}(\|x_j - x_k\|_{p_2}) \; .$$

Thus problem (PL) is restated as that of minimizing

$$f(x) + h(Ax) \qquad \text{over} \quad x \in R^{2n} \; . \tag{4.7}$$

In order to derive the Fenchel's dual to this problem, we have to calculate the conjugates of f and g , where g is a concave function on $R^{n(n-1)}$ defined by

$$g(z) = - h(z) \; .$$

Since it is easy to verify that

$$g^*(u) = - h^*(-u) \; , \tag{4.8}$$

where g^* is the concave conjugate of g and h^* is the convex conjugate of h , it suffices to calculate f^* and h^* .

It follows from Lemma 4.1 and the definitions of the conjugate and the monotone conjugate that for any $v \in R^{2n}$,

$$f_i^*(v) = \sup_x \{ \langle x,v\rangle - f_i(x) \}$$

$$= \sup_{\xi_j \geq 0} \sup_{\|x_j - a_i\|_{p_1} = \xi_j} \sum_{j=1}^{n} \{ \langle x_j, v_j\rangle - f_{ji}(\|x_j - a_i\|_{p_2}) \}$$

$$= \sum_{j=1}^{n} [\langle a_i, v_j\rangle + \sup_{\xi_j \geq 0} \sup_{\|x_j - a_i\|_{p_1} = \xi_j} \{ \langle x_j - a_i, v_j\rangle$$

$$- f_{ji}(\|x_j - a_i\|_{p_1}) \}]$$

$$= \sum_{j=1}^{n} [\langle a_i, v_j\rangle + \sup_{\xi_j \geq 0} \{ \xi_j \| v_j \|_{q_1} - f_{ji}(\xi_j) \}]$$

$$= \sum_{j=1}^{n} [\langle a_i, v_j\rangle + f_{ji}^{+}(\|v_j\|_{q_1})] ,$$

where v_j are vectors in R^2 such that $v = (v_1, \dots, v_n)^T$.

Applying [R5 , Thm.16.4], we obtain

$$f^*(v) = \inf_{v^1,\dots,v^m} \{ \sum_{i=1}^{m} f_i^*(v^i) \mid \sum_{i=1}^{m} v^i = v \}$$

$$= \inf_{v^1,\dots,v^m} \{ \sum_{i=1}^{m} \sum_{j=1}^{n} [\langle a_i, v_j^i\rangle + f_{ji}^{+}(\|v_j^i\|_{q_1})]$$

$$\mid \sum_{i=1}^{m} v^i = v \} . \tag{4.9}$$

In a similar manner, it can be shown that

$$h^*(u) = \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} g_{jk}^{+}(\|u_{jk}\|_{q_2}) . \tag{4.10}$$

It is now possible to construct the Fenchel's dual problem of (4.7) with $g = -h$. In fact, by (4.8),(4.9) and (4.10), the dual may be written as

$$\sup \{ g^*(u) - f^*(A^T u) \}$$

$$= \sup \{ - \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} g_{jk}^{+}(\|u_{jk}\|_{q_2}) - \inf_{v^1,\dots,v^m} \{ \sum_{i=1}^{m} \sum_{j=1}^{n} [\langle a_i , v_j^i \rangle + f_{ji}^{+}(\|v_j^i\|_{q_1})] \mid \sum_{i=1}^{m} v^i = A^T u \} \} ,$$

which is precisely problem (DL). It is noted that the supremum is always attained since assumption (a) of Fenchel's duality theorem [R5 , Cor.31.2.1] is obviously satisfied. On the other hand, the infimum of problem (PL) is attained since assumption (b) of [R5 , Cor.31.2.1] is satisfied. □

4.4 Examples

In this section, we shall illustrate two examples of problems whose dual problems are most effectively solved. The first example is the one which has been commonly termed as the multifacility location problem with rectangular norm and has been extensively studied by many authors. It will be shown that its dual becomes a linear programming problem. The second example seems rather uncommon in literature. But it is of interest since its dual is formulated as a quadratic programming problem.

Example 4.1. Let us consider the problem where each transportation cost is linear with respect to the distance and each distance is measured by the rectangular norm, i.e.,

$$f_{ji}(t) = \alpha_{ji} t \quad (\alpha_{ji} \geqq 0)$$

and

$$g_{jk}(t) = \beta_{jk} t \quad (\beta_{jk} \geqq 0)$$

for all i, j and k, and $p_1 = p_2 = 1$. This problem was already stated as (4.2) in Section 4.1.

It is easy to observe that the monotone conjugates of f_{ji} and g_{jk} are respectively

$$f^{+}_{ji}(s) = 0 \quad \text{if} \quad s \leqq \alpha_{ji} , \quad = +\infty \quad \text{otherwise}$$

and

$$g^{+}_{jk}(s) = 0 \quad \text{if} \quad s \leqq \beta_{jk} , \quad = +\infty \quad \text{otherwise.}$$

Since $p_1 = p_2 = 1$ implies $q_1 = q_2 = +\infty$, the dual of (4.2) may be formulated as follows:

$$\underset{u,v^1,\ldots,v^m}{\text{maximize}} \quad -\sum_{i=1}^{m}\sum_{j=1}^{n} \langle a_i, v_j^i \rangle$$

subject to

$$\sum_{i=1}^{m} v^i = A^T u \,, \tag{4.11}$$

and

$$\|u_{jk}\|_\infty \leqq \beta_{jk} \,, \quad k=j+1,\ldots,n;\ j=1,\ldots,n-1,$$

$$\|v_j^i\|_\infty \leqq \alpha_{ji} \,, \quad i=1,\ldots,m;\ j=1,\ldots,n \,.$$

Note that the constraints $\|u_{jk}\|_\infty \leqq \beta_{jk}$ and $\|v_j^i\|_\infty \leqq \alpha_{ji}$ merely determine upper and lower bounds of the variables. Hence the above problem (4.11) is a linear programming problem with 2n equality constraints and 2mn+n(n-1) bounded variables.

This dual problem (4.11) has recently been obtained by Juel and Love [J3] without explicit use of Fenchel's duality theorem. It has also been pointed out in [J3] that the optimal locations $x_1, x_2, \ldots, x_n$ for problem (4.2) are given as the optimal multipliers corresponding to the constraints $\Sigma_i v^i = A^T u$.

Example 4.2. Let the transportation cost and the measure of distances between the new facilities and the existing ones be the same as those in Example 4.1, whereas between new facilities themselves let the transportation cost be quadratic with respect to the distance measured by the Euclidean norm, i.e.,

$$f_{ji}(t) = \alpha_{ji} t \qquad (\alpha_{ji} \geqq 0)$$

and

$$g_{jk}(t) = \beta_{jk} t^2 \qquad (\beta_{jk} \geqq 0)$$

for all i, j and k and $p_1 = 1$ and $p_2 = 2$. Then problem (4.1) becomes

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{minimize}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ji} \| x_j - a_i \|_1 \\ & + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \beta_{jk} \| x_j - x_k \|_2^2 \end{aligned} \tag{4.12}$$

This problem also involves nondifferentiable functions.

Let us now consider the dual of problem (4.12). To this end, it is only necessary to calculate the monotone conjugates of g_{jk}, since the others are the same as those in Example 4.1. By the definition, it is not difficult to obtain the formula

$$g^{+}_{jk}(s) = \frac{1}{4\beta_{jk}} s^2 \quad \text{for every} \quad s \geqq 0 \quad (\beta_{jk} > 0)$$

and

$$g^{+}_{jk}(s) = 0 \quad \text{if} \quad s = 0 \;, \; = +\infty \quad \text{if} \; s > 0 \quad (\beta_{jk} = 0) \; .$$

Assuming temporarily that $\beta_{jk} > 0$ for all j and k, and noting that $q = \infty$ and $q_2 = 2$, we may derive the following dual problem:

$$\begin{aligned} \underset{u,v^1,\ldots,v^m}{\text{maximize}} \quad & - \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \frac{1}{4\beta_{jk}} \| u_{jk} \|_2^2 - \sum_{i=1}^{m}\sum_{j=1}^{n} \langle a_i, v_j^i \rangle \\ \text{subject to} \quad & \\ & \sum_{i=1}^{m} v^i = A^T u \,, \\ \text{and} \quad & \\ & \| v_j^i \|_\infty \leqq \alpha_{ji} \,, \quad i=1,\ldots,m;\; j=1,\ldots,n \,. \end{aligned} \tag{4.13}$$

If $\beta_{jk} = 0$ for some j and k, then the quadratic term $(1/4\beta_{jk})\times$

$\| u_{jk} \|_2^2$ vanishes in (4.13) and the constraint $u_{jk} = 0$ should be added to those of (4.13).

In any case, the dual problem is a quadratic programming problem with bounded variables and the objective is concave. In particular, if $\beta_{jk} > 0$ for all j and k, then problem (4.13) has a negative definite objective function and 2n linear constraints with 2mn bounded variables and n(n-1) free variables.

Consequently, we can solve problem (4.12) by solving its dual (4.13) by some efficient quadratic programming algorithm.

In a way similar to that of [J2], we obtain the Lagrangean dual of the dual problem (4.13). Define the Lagrangean $L(u,v,x)$ by

$$L(u,v,x) = -\sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \frac{1}{4\beta_{jk}} \| u_{jk} \|_2^2 - \sum_{i=1}^{m}\sum_{j=1}^{n} \langle a_i, v_j^i \rangle$$

$$+ \langle x, \sum_{i=1}^{m} v^i - A^T u \rangle ,$$

where x is interpreted as the multiplier vector. Then by a duality theorem in nonlinear programming, the objective function of the dual of (4.13) is

$$\begin{aligned} d(x) &= \max_{u,v} L(u,v,x) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \max_{v} \{ \langle v_j^i, x_j - a_i \rangle \mid \| v_j^i \|_\infty \leqq \alpha_{ji} \} \\ &\quad - \min_{u} \{ \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \frac{1}{4\beta_{jk}} \| u_{jk} \|_2^2 + \langle Ax, u \rangle \} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ji} \| x_j - a_i \|_1 + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \beta_{jk} \| -z_{jk} \|_2^2 , \end{aligned}$$

where we have used (4.3). Note that $d(x)$ is identical with the objective function of problem (4.12). Thus we can obtain the optimal locations $x_1, x_2, \ldots, x_n$ as the optimal multipliers corresponding to the constraints $\Sigma_i v^i = A^T u$ in (4.13).

4.5 Computational Results

A simple problem is solved by way of illustration. The sample problem is to find an optimal location of three new facilities among five existing facilities. The cost function is given by (4.2), where the rectangular norm is assumed, i.e., $p_1 = p_2 = 1$. The relevant data are given in Tables 4.1, 4.2 and 4.3. The optimal location of new facilities are $x_1 = (3,3)$, $x_2 = (5,4)$ and $x_3 = (5,4)$, and the cost function takes the value 45.5 at the optimum.

According to (4.11), the dual of the sample problem is stated as follows:

$$\begin{aligned}
\text{maximize}\quad & -2(v^1_{11} + v^1_{21} + v^1_{31}) - 3(v^1_{12} + v^1_{22} + v^1_{32}) \\
& -4(v^2_{11} + v^2_{21} + v^2_{31}) - 2(v^2_{12} + v^2_{22} + v^2_{32}) \\
& -5(v^3_{21} + v^3_{31}) - 4(v^3_{22} + v^3_{32}) \qquad (4.14) \\
& -3(v^4_{11} + v^4_{21}) - 5(v^4_{12} + v^4_{22}) \\
& -6(v^5_{21} + v^5_{31}) - 7(v^5_{22} + v^5_{32})
\end{aligned}$$

subject to

$$\begin{aligned}
& v^1_{11} + v^2_{11} + v^4_{11} = u_{121} , \\
& v^1_{12} + v^2_{12} + v^4_{12} = u_{122} , \\
& v^1_{21} + v^2_{21} + v^3_{21} + v^4_{21} + v^5_{21} = - u_{121} + u_{231} , \\
& v^1_{22} + v^2_{22} + v^3_{22} + v^4_{22} + v^5_{22} = - u_{122} + u_{232} , \\
& v^1_{31} + v^2_{31} + v^3_{31} + v^5_{31} = - u_{231} , \\
& v^1_{32} + v^2_{32} + v^3_{32} + v^5_{32} = - u_{232} ,
\end{aligned}$$

i	a_i
1	(2,3)
2	(4,2)
3	(5,4)
4	(3,5)
5	(6,7)

Table 4.1 a

j \ i	1	2	3	4	5
1	1.5	1	0	1	0
2	1	1	2	1	2
3	2	1	3	0	2

Table 4.2 α_{ji}

j \ k	2	3
1	1	0
2	-	1

Table 4.3 β_{jk}

and

$$-1 \leqq u_{12i} \leqq 1 \ , \ -1 \leqq u_{231} \leqq 1 \ , \ -1.5 \leqq v_{1i}^{1} \leqq 1.5 \ ,$$

$$-1 \leqq v_{2i}^{1} \leqq 1 \ , \ -2 \leqq v_{3i}^{1} \leqq 2 \ , \ -1 \leqq v_{1i}^{2} \leqq 1 \ ,$$

$$-1 \leqq v_{2i}^{2} \leqq 1 \ , \ -1 \leqq v_{3i}^{2} \leqq 1 \ , \ -2 \leqq v_{2i}^{3} \leqq 2 \ ,$$

$$-3 \leqq v_{3i}^{3} \leqq 3 \ , \ -1 \leqq v_{1i}^{4} \leqq 1 \ , \ -1 \leqq v_{2i}^{4} \leqq 1 \ ,$$

$$-2 \leqq v_{2i}^{5} \leqq 2 \ , \ -2 \leqq v_{3i}^{5} \leqq 2, \quad i = 1, 2 \ .$$

The computation time for the problem (4.14) (including compilation) was about 4 seconds on a FACOM M-190 computer of Kyoto University Computer Center.

It may be noted that this sample problem was first considered by Wesolowsky and Love [W1], where they solved the dual of a linear programming problem derived from the original problem by introducing artificial variables. By comparison, the number of variables and that of constraints of (4.14) are just the same as those in [W1], though the coefficients are different. As was expected, the computation time for solving (4.14) was almost the same as that for solving the dual in [W1].

4.6 Conclusion

Throughout this chapter, we have restricted our attention to problems in R^2, because most problems seem to have been considered on a plane. However, it can be easily verified that the duality of problems (PL) and (DL) still remains valid in any dimensional Euclidean spaces.

Finally, we mention that the dual problem (DL) includes that of Juel and Love [J3] as a special case, as far as unconstrained problems are concerned. Furthermore, it is expected that the approach proposed in this chapter may be extended to a certain class of constrained generalized multifacility location problems as well.

CHAPTER 5

MINIMIZATION OF THE SUM OF A CONVEX FUNCTION AND A CONTINUOUSLY DIFFERENTIABLE FUNCTION

In this chapter, we present a method of finding the minimum for a class of nonconvex and nondifferentiable functions consisting of the sum of a convex function and a continuously differentiable function. The class of such problems is general enough to cover certain constrained minimization problems. The algorithm proposed in this chapter is a descent method which generates successive search directions by solving auxiliary minimization problems which consists of convex functions. It is shown that the algorithm generates a sequence of points converging to a critical point of the problem.

5.1 Introduction

Recently, considerable attention has been paid to minimization problems without the differentiability assumptions on the functions involved. In particular, if the function to be minimized is convex, various algorithms for nondifferentiable convex optimization problems may be applied [B11][L2][P5][S2][W6]. Most of these methods utilize the subgradients or ε-subgradients [R5] of convex functions. More recently, Feuer [F3][F4], Goldstein [G8] and Mifflin [M4] have used the Clarke generalized gradients [C3] to extend these techniques to apply to various classes of nonconvex and nondifferentiable functions.

In this chapter, we shall concentrate our attention to a class of nonconvex and nondifferentiable functions consisting of the sum of a convex function and a continuously differentiable function, and propose a method of finding the minima of these functions. Such problems contain a fairly wide class of problems which are encountered in practice and indeed involve some important class of constrained minimization problems. It may be noted that such functions are semismooth, good, semiconvex, quasi-differentiable in the sense of Mifflin [M5], Feuer [F4], Vainberg [V1], Pshenichnyi [P8], respectively, and Clarke differentiable [C3], if they are finite everywhere.

The algorithm under consideration here is a descent method generating successive search directions by solving successive convex auxiliary problems, each of which may be regarded as an approximation of the original problem. The present algorithm is particularly useful if the convex function involved is simple enough that it can be minimized without the aid of general purpose convex minimization algorithms such as those in [L2][P5][S2][W6], as illustrated in Examples 5.1 and 5.3 below. Example 5.1 also suggests that the present algorithm may be regarded as an natural extension of the Frank-Wolfe algorithm [F12] for constrained minimization problems.

This chapter is organized as follows: In Section 5.2, the minimization problem is stated and some examples are given to indicate the possibility of reducing constrained minimization problems into the problem considered in this chapter. In Section 5.3, the optimality condition for the problem is given and the algorithm is formally described. In Section 5.4, convergence properties of the algorithm are proved. Section 5.5 discusses the convergence rate of the algorithm for quadratic functions and shows a relationship to steepest descent and conjugate directions.

5.2 The Problem

The problem considered in this chapter is

$$\text{minimize} \quad \phi(x) \triangleq f(x) + g(x) \qquad \text{over} \quad x \in R^n , \tag{5.1}$$

where $g: R^n \to (-\infty, +\infty]$ is a closed proper convex, but not necessarily differentiable function and $f: R^n \to (-\infty, +\infty]$ is a function which is continuously differentiable on an open set containing $\operatorname{dom} g$, but f need not be convex. The function ϕ is thus in general neither convex nor differentiable.

Although we are mainly concerned with the unconstrained problem (5.1), the present method can be applied to constrained problems as shown in the following examples.

Example 5.1. Consider the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in S \subset R^n , \end{aligned} \tag{5.2}$$

where $f: R^n \to (-\infty, +\infty)$ is a continuously differentiable function and S is a closed convex set. Then, problem (5.2) may be rewritten in an equivalent form

$$\text{minimize} \quad f(x) + \delta_S(x) \qquad \text{over} \quad x \in R^n , \tag{5.3}$$

where δ_S is the indicator function of S [R5] defined by

$$\begin{aligned} \delta_S(x) &= 0 && \text{if} \quad x \in S \\ &= +\infty && \text{otherwise.} \end{aligned}$$

Clearly, problem (5.3) is the problem of the form (5.1).

Example 5.2. For an inequality constrained problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && h_i(x) \leqq 0 \qquad i=1,\dots,m , \end{aligned} \tag{5.4}$$

where $h_i: R^n \to (-\infty, +\infty)$ are convex functions and f is the same as that in Example 5.1, we may construct a penalty function

$$f(x) + \sum_{i=1}^{m} p_i[h_i(x)] , \tag{5.5}$$

where $p_i: R^n \to (-\infty, +\infty)$ are convex functions such that

$$\begin{aligned} p_i(t) &> 0 \quad \text{for} \quad t > 0 \\ &= 0 \quad \text{for} \quad t \leqq 0 . \end{aligned}$$

Then it is known [B8][C6][E3][H6][P1][Z1] that, under certain assumptions, the penalty function (5.5) is exact in the sense that an unconstrained minimum of (5.5) is an optimal solution of the constrained problem (5.4). Therefore, problem (5.4) may be rewritten as

$$\text{minimize} \quad f(x) + \sum_{i=1}^{m} p_i[h_i(x)] \quad \text{over} \quad x \in R^n .$$

Since the second term is convex in x, problem (5.4) can be transformed into an unconstrained problem of the form (5.1).

Remark 5.1. For a given problem, decomposition of ϕ into a differentiable function f and a convex function g is not unique, since ϕ is not altered by subtracting any continuously differentiable convex function from f and adding the same function to g. However, convergence rate of the present algorithm is dependent on a manner of decomposition of ϕ into f and g. This subject will be discussed in Section 5.5 for a simple class of problems.

5.3 Algorithm

In this section, we shall state the algorithm for finding a critical point of problem (5.1). First we consider a necessary condition for a point x to be an optimal solution of (5.1). Clarke [C4] has shown that if a point x is a local minimum of a locally Lipschitz function ψ (the function ψ is locally Lipschitz if for any x there is a neighborhood of x such that, for some constant K ,

$$|\psi(y) - \psi(z)| \leqq K \| y-z \|$$

for any y and z), then $0 \in \partial^*\psi(x)$, where $\partial^*\psi(x)$ is the *Clarke generalized gradient* of ψ at x .

If g is finite everywhere on R^n , then ϕ is locally Lipschitz and $\partial^*\phi(x) = \nabla f(x) + \partial g(x)$, where ∂g is the *subdifferential* of g at x . Hence, a necessary condition for a point x to be a local minimum of ϕ is $-\nabla f(x) \in \partial g(x)$. For the function g which may take the value $+\infty$ somewhere, this result is not directly applicable. However, under a certain assumption, we can show that this is valid for extended-real-valued functions as well.

Throughout this paper, we shall make the following assumption: For any $x \in \operatorname{dom} g$ and any d ,

$$g'(x;d) = \liminf_{e \to d} g'(x;e) ,$$

where $g'(x;d)$ is the *one-sided directional derivative* of g at x with respect to the direction y . Under this assumption, we have

$$g'(x;d) = \sup \{ \langle y,d \rangle \; ; \; y \in \partial g(x) \}$$

for any $x \in \operatorname{dom} g$ and any d , by [R5 , Thm. 23.2]. It is noted that, if $x \in \text{ri dom } g$, or in particular if $\text{dom } g = R^n$, then the above equality is valid [R5 , Thm.23.4].

Then the following proposition is proved.

Proposition 5.1. A point x is a local minimum of problem (5.1) only if $-\nabla f(x) \in \partial g(x)$.

Proof. Suppose that x is a local minimum of problem (5.1). Clearly, $x \in \text{dom } g$. Then, for any direction d , the one-sided directional derivative of ϕ at x is

$$\begin{aligned}\phi'(x;d) &= f'(x;d) + g'(x;d) \\ &= \langle\nabla f(x),d\rangle + \sup\{ \langle y,d\rangle \; ; \; y \in \partial g(x) \} \\ &= \sup\{ \langle\nabla f(x)+y,d\rangle \; ; \; y \in \partial g(x) \} .\end{aligned}$$

On the other hand, since x is a local minimum of ϕ , we have

$$\phi'(x;d) \geqq 0 \qquad \text{for any } d .$$

Therefore, by a separation theorem, it is not difficult to see that

$$0 \in \nabla f(x) + \partial g(x) . \qquad \square$$

In what follows, any point x which satisfies the condition in the above proposition is called a *critical point* of problem (5.1). Obviously, the condition is sufficient when ϕ is actually convex or semiconvex as defined [M5].

There are already a number of algorithms for the minimization of convex functions [B11][L2][P5][S2][W6]. This chapter proposes an iterative algorithm for generating critical points of functions which are "almost convex" in the sense that they may be expressed in the form of a sum of a convex function g and a continuously differentiable function f. The key to the algorithm is that successive descent directions from the point x_k may be selected to be $\tilde{x} - x_k$ where $\tilde{x}$ is a minimizing solution of the convex problem

$$\text{minimize} \quad \langle x, \nabla f(x_k)\rangle + g(x) \qquad \text{over} \quad x \in R^n . \tag{5.6}$$

An algorithm may then be defined to generate critical points of ϕ as follows:

<u>Algorithm</u>.

Step 1: Let x_0 be any initial point. Set $k = 0$ and go to step 2.

Step 2: If $-\nabla f(x_k) \in \partial g(x_k)$, stop. Otherwise, go to step 3.

Step 3: Find a minimum $\tilde{x}_k$ of problem (5.6), and go to step 4.

Step 4: Find $x_{k+1} = x_k + \lambda_k d_k$ such that $\lambda_k \geqq 0$ and

$$\phi(x_{k+1}) \leqq \phi(x_k+\lambda d_k) \qquad \text{for all} \quad \lambda \geqq 0 ,$$

where $d_k = \tilde{x}_k - x_k$. Set $k = k+1$ and go to step 2.

It is important to recognize that this method permits one to take advantage of the special structure of the objective function ϕ. In particular it permits the efficient optimization of objectives which may be decomposed into the sum of an easy-to-optimize convex function and a continuously differentiable function. Thus the proposed algorithm differs significantly from the general purpose algorithms which were proposed by Feuer [F3][F4] and Mifflin [M4].

<u>Remark 5.2</u>. The stopping criterion in step 2 may be replaced by

"if $d_k = 0$, stop." or "if $\|d_k\| < \varepsilon$, stop.", where d_k is determined in step 4 and ε is some prescribed small positive number. The validity of these alternatives may be verified by using Lemma 3.1 which will be given in the next section.

Remark 5.3. It should be noted that solving (5.6) in step 3 is equivalent to finding $\tilde{x}_k$ such that $-\nabla f(x_k) \in \partial g(\tilde{x}_k)$. The latter is again equivalent to $\tilde{x}_k \in \partial g^*(-\nabla f(x_k))$ by [R5, Thm.23.5], where g^* is the conjugate function of g [R5, p.104]. So it will be very easy to obtain $\tilde{x}_k$ directly from $\partial g^*(-\nabla f(x_k))$, when ∂g^* has a convenient characterization. Unfortunately, the class of such functions is somewhat limited.

Remark 5.4. For problem (5.2) in Example 5.1, the algorithm is the same as the Frank-Wolfe algorithm [F12] for quadratic programming or a more general conditional gradient method [L3] except for the global line search in step 4. It is thus suggested that finite line search techniques or such a simple step-size rule as in [L3] may be employed instead of the exact one used in the proposed algorithm.

It should be noted that there exists a minimum $\tilde{x}_k$ of problem (5.6) if and only if $-\nabla f(x_k) \in \text{range}\ \partial g$. In what follows, however, we shall make a stronger assumption that $\text{range}(-\nabla f) \subset \text{int range}\ \partial g$, or equivalently that $\text{range}(-\nabla f) \subset \text{int dom}\ \partial g^* = \text{int dom}\ g^*$. The last condition is satisfied in particular if g is co-finite [R5 , p.116], since the co-finiteness of g implies that $\text{dom}\ g^* = R^n$ [R5 , Cor.13.3.1].

5.4 Convergence of Algorithm

The following theorem shows the algorithm converges.

<u>Theorem 5.1.</u> Let x_0 be an arbitrary initial point. Assume that the level set $L(x_0) = \{ x \quad R^n \; ; \; \phi(x) \leqq \phi(x_0) \}$ is compact and that g is strictly convex on $\operatorname{dom} g$. Then the sequence $\{x_k\}$ generated by the algorithm contains a subsequence which converges to a critical point of problem (5.1).

To prove the theorem, it will be convenient to establish several lemmas. The first lemma gives a condition for a point x_k to be a critical point.

<u>Lemma 5.1.</u> If $d_k = \tilde{x}_k - x_k = 0$, then x_k is a critical point of (5.1). Moreover, the converse is true provided that g is strictly convex on $\operatorname{dom} g$.

<u>Proof.</u> First let us suppose that x_k is not a critical point. Then $-\nabla f(x_k) \notin \partial g(x_k)$ which is equivalent to $x_k \notin \partial g^*(-\nabla f(x_k))$ by [R5 , Thm. 23.5], while $\tilde{x}_k$ is chosen by definition so that $\tilde{x}_k \in \partial g^*(-\nabla f(x_k))$. Thus $\tilde{x}_k$ cannot be identical with x_k .

Now assume that g is strictly convex on $\operatorname{dom} g$ and that x_k is a critical point. Then $-\nabla f(x_k) \in \partial g(x_k)$. But, since $\tilde{x}_k$ satisfying $-\nabla f(x_k) \in \partial g(\tilde{x}_k)$ is uniquely determined by the strict convexity of g , x_k must coincide with $\tilde{x}_k$. □

<u>Remark 5.5.</u> If g is strictly convex on $\operatorname{dom} g$, then each subproblem (5.6) achieves its minimum uniquely. Namely, $\tilde{x}_k$ is uniquely

determined in step 3 of the algorithm and is given by $\nabla g^*(-\nabla f(x_k))$, since the strict convexity of g on dom g implies that g is essentially strictly convex [R5 , p.253], which in turn implies that g^* is essentially smooth [R5 , Thm.26.3], i.e., g^* is differentiable on $\text{range}(-\nabla f) \subset \text{int dom } g^*$.

The next lemma shows that d_k is a direction of descent of ϕ at x_k .

Lemma 5.2. For any x_k and $d_k = \tilde{x}_k - x_k$,

$$\phi'(x_k;d_k) \leqq 0 .$$

Moreover, the inequality is strict, provided that x_k is not a critical point and that g is strictly convex on dom g .

Proof. We first note that the one-sided directional derivative $\phi'(x_k;d_k)$ is given by

$$\begin{aligned}
\phi'(x_k;d_k) &= \lim_{\lambda\downarrow 0} \lambda^{-1}[\ \phi(x_k+\lambda d_k) - \phi(x_k)\ ] \\
&= \lim_{\lambda\downarrow 0} \lambda^{-1}[\ f(x_k+\lambda d_k) - f(x_k)\ ] \\
&\quad + \lim_{\lambda\downarrow 0} \lambda^{-1}[\ g(x_k+\lambda d_k) - g(x_k)\ ] \\
&= \langle\nabla f(x_k), d_k\rangle + \sup\{\ \langle y, d_k\rangle\ ;\ y \in \partial g(x_k)\ \} \\
&\qquad\qquad (\ [\text{ R5 , Thm.23.4}]\ ) \\
&= \sup\{\ \langle\nabla f(x_k)+y, d_k\rangle\ ;\ y \in \partial g(x_k)\ \} . \qquad (5.7)
\end{aligned}$$

By the monotonicity of the mapping ∂g [R5 , Cor.31.5.2], we have

$$\langle-\nabla f(x_k)-y, \tilde{x}_k-x_k\rangle \geqq 0 \qquad \text{for any } y \in \partial g(x_k) , \qquad (5.8)$$

since $-\nabla f(x_k) \in \partial g(\tilde{x}_k)$. It can also be shown that the inequality is strict if g is strictly convex and $x_k \neq \tilde{x}_k$ which is true by Lemma 5.1 when x_k is not a critical point. Since $d_k = \tilde{x}_k - x_k$, the lemma follows from (5.7) and (5.8). □

Let us now consider the algorithmic point-to-set map $A: R^n \to R^n$ associated with the algorithm in a composite form

$$A = MD ,$$

where the map D determines a direction d_k at a current point x_k and the map M finds a minimum point x_{k+1} along the direction d_k . Specifically, by Remark 5.3, the map $D: R^n \to R^{2n}$ is defined by

$$D(x) = \{ (x,d) \; ; \; d = \tilde{x} - x \quad \text{for some } \tilde{x} \quad \text{such that } -\nabla f(x) \in \partial g(\tilde{x}) \}$$

and the map $M: R^{2n} \to R^n$ is defined by

$$M(x,d) = \{ x^+ \; ; \; x^+ = x + \lambda d \quad \text{for some } \lambda \geqq 0 \quad \text{such that } \phi(x^+) = \min_{\lambda \geqq 0} \phi(x+\lambda d) \} .$$

Recall that the map A is said to be *closed* at x [Z2, p.88] if

$$x_k \to x \quad (k \to \infty) ,$$

$$y_k \in A(x_k) , \quad k=1,2,\ldots ,$$

and

$$y_k \to y \quad (k \to \infty)$$

imply

$$y \in A(x) .$$

Lemma 5.3. The algorithmic map A is closed at any point x which is not a critical point of problem (5.1).

Proof. First let us show the closedness of D. Let us write D in the form

$$D = D_3 D_2 D_1 ,$$

where

$$D_1(x) = \{ (x,y) \; ; \; y = -\nabla f(x) \} ,$$

$$D_2(x,y) = \{ (x,\tilde{x}) \; ; \; y \in \partial g(\tilde{x}) \} ,$$

and

$$D_3(x,\tilde{x}) = \{ (x,d) \; ; \; d = \tilde{x} - x \} .$$

Clearly, $D_1 \colon R^n \to R^{2n}$ and $D_3 \colon R^{2n} \to R^{2n}$ are continuous point-to-point maps.

Now let us show that the map $D_2 \colon R^{2n} \to R^{2n}$ is closed at every point (x,y) such that $y = -\nabla f(x)$. Indeed it is closed since $y \in \partial g(x)$ if and only if $x \in \partial g^*(y)$ [R5, Cor.23.5.1] and ∂g^* is closed on range$(-\nabla f)$ [R5, Thm.24.4]. Moreover, since ∂g^* is locally uniformly bounded at any point $y \in$ range$(-\nabla f) \subset$ int dom g^* [R5, Thm.24.7], the map D_2 has the following property:

(p) if $(x_i,y_i) \to (x,y)$ and $(x_i,\tilde{x}_i) \in D_2(x_i,y_i)$, then there exists some $\tilde{x}$ such that, for some subsequence $\{(x_{i_\ell},\tilde{x}_{i_\ell})\}$, $(x_{i_\ell},\tilde{x}_{i_\ell}) \to (x,\tilde{x})$.

Then it follows from [Z2, Lemma 4.2] that the composite map $D_3 D_2$ is closed. Moreover, by [Z2, Cor.4.2.2], it is easy to see that the map $D = D_3 D_2 D_1$ is also closed. It is also easily seen that the map D has a property similar to (p) above.

On the other hand, it is known [L8 , p.146] that the map M is closed at (x,d) whenever $d \neq 0$. Thus by [Z2 , Lemma 4.2] and by Lemma 3.1, we may conclude that the map A is closed at any x which is not a critical point. $\square$

<u>Remark 5.6.</u> If we had assumed that g was strictly convex as we did for Theorem 5.1, the proof of Lemma 5.3 could have been considerably simplified. The map D_2 would then have reduced to a point-to-point map which would be continuous at all (x,y) for which $y = -\nabla f(x)$. We however preferred to prove Lemma 5.3 under more general conditions.

We are now ready to prove Theorem 5.1.

<u>Proof of Theorem 5.1.</u> (i) Obviously, the entire sequence $\{x_k\}$ lies in the compact level set $L(x_0)$ by the descent property of the algorithm.

(ii) The function ϕ is continuous on $L(x_0)$.

(iii) (a) If x is not a critical point, then by the strict convexity of g and by Lemma 5.2, $\phi'(x;d) < 0$ where d is determined by the map D at x . So $\phi(x) > \phi(x^+)$ for any $x^+ \in A(x)$.

(b) If x is a critical point, then the algorithm terminates.

(iv) The map A is closed at x if x is not a critical point by Lemma 5.3.

Hence, by the well-known Convergence Theorem A of Zangwill [Z2 , p.91], it follows that there exists a subsequence of $\{x_k\}$ converging to a critical point of problem (5.1). This completes the proof. □

Remark 5.7. For problem (5.2), convergence of the algorithm may be proved under different assumptions such as the convexity of f (see [L3 , Thm.6.1].)

We have proved convergence of the algorithm under the assumption of strict convexity of g . It might seem that the algorithm is not applicable to many important classes of problems which do not satisfy this assumption, such as problem (5.2) in Example 5.1. It is to be noted, however, that we can always *strictly convexify* g without changing problem itself by adding a continuously differentiable strictly convex function to g and subtracting the same function from f . A typical case is examined in Example 5.3 below.

Example 5.3. Consider the following problem:

$$\begin{array}{ll} \text{minimize} & f(x) + h(x) \\ \text{subject to} & x \in S , \end{array} \qquad (5.9)$$

where f is continuously differentiable on R^n , S is a polyhedral convex set and h is given by

$$h(x) = \max \{ \langle a_i, x\rangle ; i=1,\ldots,m \} .$$

Let

$$g(x) = h(x) + \delta_S(x) ,$$

where δ_S is the indicator function of S. Then g is not strictly convex on dom g. However, we can modify this problem so as to satisfy the assumption of strict convexity. In fact, defining

$$\tilde{f}(x) = f(x) - \tfrac{1}{2} \langle x, Ax \rangle$$

and

$$\tilde{g}(x) = g(x) + \tfrac{1}{2} \langle x, Ax \rangle ,$$

where A is an arbitrary symmetric positive definite matrix, we can easily see that $\tilde{g}$ is strictly convex on dom g and that problem (5.9) is equivalent to the following

$$\text{minimize} \quad \tilde{f}(x) + \tilde{g}(x) \qquad \text{over} \quad x \in R^n .$$

Then subproblem (5.6) becomes

$$\text{minimize} \quad \langle \nabla f(x_k) - Ax_k, x \rangle + \tilde{g}(x) \qquad \text{over} \quad x \in R^n ,$$

which in turn can be rewritten as

$$\begin{aligned} &\text{minimize} && \langle \nabla f(x_k) - Ax_k, x \rangle + t + \tfrac{1}{2} \langle x, Ax \rangle \\ &\text{subject to} && \langle a_i, x \rangle \leqq t , \quad i=1,\ldots,m , \\ &\text{and} && x \in S , \end{aligned} \tag{5.10}$$

where t is an additional variable. Each subproblem (5.10) may be sequentially solved by some parametric quadratic programming method, since only coefficients of the linear part of the objective function vary at each iteration.

Note that direct application of the algorithm to problem (5.9) yields the following linear programming subproblem instead of (5.10)

$$\begin{aligned} &\text{minimize} && \langle \nabla f(x_k), x \rangle + t \\ &\text{subject to} && \langle a_i, x \rangle \leqq t , \quad i=1,\ldots,m , \\ &\text{and} && x \in S . \end{aligned} \tag{5.11}$$

It should be noted that this problem contains the Frank-Wolfe subproblem as a special case, i.e., $m=0$.

Let us now mention how the algorithm would process a function ϕ for which (a) $g(x) = 0$, i.e., ϕ is continuously differentiable and (b) $f(x) = 0$, i.e., ϕ is convex.

In case (a), the algorithm will not work since subproblem (5.6) does not have a solution in general. However, we may still apply the algorithm by setting

$$\phi(x) = \{ f(x) - \tfrac{1}{2} \langle x,x \rangle \} + \tfrac{1}{2} \langle x,x \rangle$$

as in Example 5.3. Then the algorithm would behave like a steepest descent method for ϕ (see Section 5.5.)

For case (b), the algorithm terminates in a single step. However, we can still take advantage of the structure of g . For example, consider the problem (5.9) in Example 5.3 and suppose the objective function $f + h$ is convex. Then this is exactly case (b). Although (5.9) may be solved by a general purpose convex minimization algorithm such as [L2] [P5] [S2] [W6], it will be sometimes preferable to employ the present algorithm since subroutines for solving subproblems (5.10) or (5.11) are widely available.

5.5 Rate of Convergence

In this section, we discuss the convergence rate of the algorithm for quadratic functions and show relations to the steepest descent method.

Recall that, in order for the algorithm to be implemented efficiently, ϕ must be decomposed so that g is an easy-to-optimize function. However, except for the trivial case that ϕ itself is an easy-to-optimize convex function, in which case the algorithm will terminate in a single step, there is flexibility in decomposing ϕ into f and g (see Examples 5.1 and 5.3.) The analysis of this section for quadratic functions shows that the decomposition scheme actually affects the convergence rate and suggests that convergence may be accelerated by an appropriate choice of decomposition scheme.

First consider a simple case where $g(x) = (\beta/2)\langle x,x\rangle$, i.e.,

$$\phi(x) = f(x) + (\beta/2)\langle x,x\rangle \quad ,$$

where β is a positive constant. Then for a given x_k , the solution of subproblem (5.6) is

$$\tilde{x}_k = -\beta^{-1}\nabla f(x_k) \quad .$$

Therefore, the direction $d_k = \tilde{x}_k - x_k$ reduces to

$$\begin{aligned} d_k &= -\beta^{-1} \{ \nabla f(x_k) + \beta x_k \} \\ &= -\beta^{-1}\nabla\phi(x_k) \quad . \end{aligned}$$

So, in this case, the algorithm coincides with the steepest descent method.

Next, we consider the case where both f and g are positive definite quadratic functions. Namely, suppose that

$$f(x) = \tfrac{1}{2} \langle x, Ax \rangle + \langle c, x \rangle$$

and

$$g(x) = \tfrac{1}{2} \langle x, Bx \rangle \qquad (5.12)$$

so that

$$\phi(x) = \tfrac{1}{2} \langle x, (A+B)x \rangle + \langle c, x \rangle$$

where B and $A+B$ are both symmetric positive definite matrices.

Since we have $\nabla f(x) = Ax + c$ and $\nabla g(x) = Bx$, step 3 of the algorithm determines $\tilde{x}_k$ for a given x_k by the formula

$$-\nabla f(x_k) = -(Ax_k + c) = B\tilde{x}_k = \nabla g(\tilde{x}_k) ,$$

or

$$\tilde{x}_k = -B^{-1}[Ax_k + c] .$$

Thus the direction d_k is given by

$$\begin{aligned} d_k &= \tilde{x}_k - x_k \\ &= -B^{-1}[(A+B)x_k + c] . \end{aligned} \qquad (5.13)$$

Now consider a one-dimensional problem

$$\underset{\lambda \geqq 0}{\text{minimize}} \ \phi(x_k + \lambda d_k) = \tfrac{1}{2} \lambda^2 \langle d_k, (A+B)d_k \rangle + \lambda \langle d_k, (A+B)x_k + c \rangle + \tfrac{1}{2} \langle x_k, (A+B)x_k + c \rangle , \qquad (5.14)$$

where d_k is given by (5.13). It is readily shown that the minimum in (5.14) is attained by

$$\begin{aligned} &= - \langle d_k, (A+B)x_k + c \rangle / \langle d_k, (A+B)d_k \rangle \\ &= \langle d_k, Bd_k \rangle / \langle d_k, (A+B)d_k \rangle \qquad \text{(by (5.13).)} \end{aligned}$$

Thus we have

$$x_{k+1} = x_k + \frac{<d_k, Bd_k>}{<d_k, (A+B)d_k>} d_k . \tag{5.15}$$

It might be convenient to introduce the function

$$E(x) = \tfrac{1}{2} < x - x^*, (A+B)(x - x^*)> ,$$

where $x^* = -(A+B)^{-1}c$ is the unique minimum of ϕ . Clearly, E differs from ϕ only by a constant $\phi(x^*)$, so that the sequence $\{x_k\}$ generated by the algorithm applied to ϕ is the same as that for E .

We have already seen that if $B = \beta I$ in (5.12), then the algorithm reduces to the steepest descent method. In general, it is known [L8 , p.152] that the convergence rate for function values of steepest descent applied to (5.12) is $[(\bar{M}-\bar{m})/(\bar{M}+\bar{m})]^2$, where $\bar{m}$ and $\bar{M}$ are the smallest and largest eigenvalues of $A+B$, respectively. In other words, for a sequence $\{x_k\}$ generated by the steepest descent method, we have

$$E(x_{k+1}) \leqq \left[\frac{\bar{M} - \bar{m}}{\bar{M} + \bar{m}} \right]^2 E(x_k) .$$

For the present algorithm, we obtain the following result.

<u>Theorem 5.2</u>. Suppose that the functions f , g and ϕ are the quadratic functions defined by (5.12). Then, any sequence $\{x_k\}$ generated by the algorithm converges to the unique solution x^* and satisfies

$$E(x_{k+1}) \leqq \left[\frac{M - m}{M + m} \right]^2 E(x_k) ,$$

where m and M are, respectively, the smallest and largest eigenvalues of $\underset{\sim}{Q} \triangleq B^{-1/2}(A+B)B^{-1/2}$.

Proof. By direct calculation, we have

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} = \frac{\langle d_k, Bd_k\rangle^2}{\langle Bd_k, (A+B)^{-1}Bd_k\rangle \langle d_k, (A+B)d_k\rangle} .$$

Setting $Q = B^{-1/2}(A+B)B^{-1/2}$ and $s = B^{1/2}d_k$, we may rewrite the above equality as follows:

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} = \frac{\langle s,s\rangle^2}{\langle s,\underset{\sim}{Q}^{-1}s\rangle \langle s,\underset{\sim}{Q}s\rangle} .$$

we may then use the Kantorovich inequality [L8, p.151] to obtain

$$\frac{E(x_k) - E(x_{k+1})}{E(x_k)} \leqq \frac{4mM}{(m+M)^2} ,$$

where m and M are the smallest and largest eigenvalues of $\underset{\sim}{Q}$, respectively. From this, the theorem follows immediately. □

Let us briefly mention a relation of the algorithm to the method of steepest descent. Let ℓ and L be, respectively, the smallest and largest eigenvalues of B and let m and M be, respectively, the smallest and largest eigenvalues of $(A+B)$ as before. Let $\rho(\cdot)$ denote the spectral radius of a matrix, i.e.,

$$\rho(\cdot) = \max_i |\lambda_i| ,$$

where λ_i are eigenvalues of the matrix [H5, p.3]. Since $A+B$ and B are positive definite, we have

$$\bar{M} = \rho(A+B) ,$$

$$\bar{m} = 1/\rho([A+B]^{-1}) ,$$

$$L = \rho(B)$$

and

$$\ell = 1/\rho(B^{-1}) .$$

By [H5, p.46], the following inequalities are satisfied

$$\rho(\underset{\sim}{Q}) \leqq \rho(B^{-1/2}) \; \rho(A+B) \; \rho(B^{-1/2})$$

and

$$\rho(\underset{\sim}{Q}^{-1}) \leqq \rho(B^{1/2}) \; \rho([A+B]^{-1}) \; \rho(B^{1/2}) ,$$

which imply

$$M \leqq \ell^{-1/2} \; \bar{M} \; \ell^{-1/2} = \bar{M}/\ell$$

and

$$1/m \leqq L^{1/2} \; (1/\bar{m}) \; L^{1/2} = L/\bar{m} .$$

Thus we obtain a bound

$$M/m \leqq (\bar{M}/\bar{m})(L/\ell) .$$

In other words, the condition number of $\underset{\sim}{Q}$ is bounded by that of $A+B$ multiplied by that of B.

Finally, we show that successive directions d_k and d_{k+1} are conjugate with respect to B, rather than $A+B$. In fact, since

$$\begin{aligned} d_{k+1} &= - B^{-1}[(A+B)x_{k+1} + c] \\ &= - B^{-1}[(A+B)x_k + c + \frac{\langle d_k, Bd_k \rangle}{\langle d_k, (A+B)d_k \rangle} (A+B)d_k] && \text{(by (5.15))} \\ &= d_k - \frac{\langle d_k, Bd_k \rangle}{\langle d_k, (A+B)d_k \rangle} B^{-1}(A+B)d_k && \text{(by (5.13)),} \end{aligned}$$

we have a conjugacy relation

$$\langle d_k, Bd_{k+1}\rangle = \langle d_k, Bd_k\rangle - \frac{\langle d_k, Bd_k\rangle}{\langle d_k, (A+B)d_k\rangle} \langle d_k, (A+B)d_k\rangle$$

$$= 0 .$$

5.6 Conclusion

We have proposed a minimization algorithm efficiently applicable to nondifferentiable and nonconvex functions which are decomposed into sums of continuously differentiable functions and easy-to-minimize convex functions. From a viewpoint of practical implementation, however, an exact line search procedure employed in the proposed algorithm might be infeasible in a strict sense since an infinite number of iterations are usually required to perform such a line search. The problem considered in this chapter will be further investigated in the next chapter and another algorithm which incorporates a line search procedure having the property of finite termination will be proposed.

CHAPTER 6

GENERALIZATION OF THE PROXIMAL POINT ALGORITHM TO CERTAIN NONCONVEX PROGRAMS

In this chapter, we deal with the problem of minimizing a function which is the sum of a continuously differentiable function and a convex function. In the previous chapter, the same problem has been discussed and a method has been proposed to solve such problems. The method presented in this chapter is closely related to the previous method and may be considered as a generalization of the proximal point algorithm to cope with nonconvexity of the objective function by linearizing the differentiable term at each iteration. Convergence of the algorithm is proved and the rate of convergence is analyzed in detail.

6.1 Introduction

Very recently, Rockafellar [R8][R11] has applied the theory of the proximal point algorithm for maximal monotone operators to convex minimization problems, and obtained some theoretical results on the convergence of the proximal minimization algorithm. The algorithm is useful, however, only for convex problems, because the idea underlying his results is based on the monotonicity of subdifferential operators of convex functions.

In this chapter, we present a method of minimizing certain nonconvex functions which are the sums of continuously differentiable functions and convex functions by generalizing the proximal point algorithm, and discuss the convergence properties of the proposed algorithm. It will be shown that the present algorithm converges with a linear convergence rate to a critical point of the problem under certain conditions. Remarkable advantage of the present algorithm is in that the convergence may be assured even without the strict convexity assumption on the function involved and that a finite procedure may be used in the line search step of the algorithm.

This chapter is organized as follows: In Section 6.2, the problem is formally stated and some background of the present algorithm is given. The algorithm is described in Section 6.3. In Section 6.4, under certain assumptions, convergence of the algorithm is proved. In Section 6.5, the convergence rate of the algorithm is analysed. Section 6.6 discusses the relations of the proposed algorithm with the proximal point algorithm, the steepest descent algorithm and the subgradient algorithm.

6.2 Preliminaries

The problem to be considered is

$$\text{minimize} \quad \phi(x) \triangleq f(x) + g(x) \quad \text{over} \quad x \in R^n , \tag{6.1}$$

where $g: R^n \to (-\infty, +\infty]$ is a closed proper convex function and $f: R^n \to (-\infty, +\infty]$ is a function which is continuously differentiable on an open set containing $\operatorname{dom} g$. The objective function ϕ is thus in general neither convex nor differentiable.

For problem (6.1), a point x is called a *critical point* if it satisfies

$$-\nabla f(x) \in \partial g(x) . \tag{6.2}$$

As is pointed out in Chapter 5, the following constrained problem may be formulated as a problem of the form (6.1):

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in C , \end{aligned}$$

where f is continuously differentiable and C is a nonempty closed convex set. For this problem, the condition (6.2) reduces to a fundamental optimality condition that $-\nabla f(x)$ is normal to C at x, since the set $\partial\delta_C(x)$ is just the normal cone to C at x [R5, Thm.27.4].

The next proposition gives a characterization of the critical point of problem (6.1).

<u>Proposition 6.1.</u> A point x is a critical point of problem (6.1) if and only if

$$\phi'(x;d) \geqq 0 \qquad \text{for all} \quad d\ , \tag{6.3}$$

where $\phi'(x,d)$ is the one-sided directional derivative of ϕ at x in the direction d. In particular, if x is a local minimum of (6.1), then x is a critical point of (6.1).

<u>Proof</u>. Suppose that x satisfies (6.2). Then by the defining inequality of subgradients of convex functions, we have

$$g(x+\lambda d) \geqq g(x) - \lambda <\nabla f(x),d>\ ,$$

for any λ and d. On the other hand,

$$f(x+\lambda d) = f(x) + \lambda <\nabla f(x),d> + o(\lambda)\ ,$$

where $o(\lambda)/\lambda \to 0$ as $\lambda \to 0$. Combining the above inequality and equality, we have for $\lambda > 0$

$$[\ \phi(x+\lambda d) - \phi(x)\ ]\ /\ \lambda \geqq o(\lambda)/\lambda\ .$$

Taking a limit as $\lambda \to 0^+$, we obtain the inequality (6.3).

Now suppose (6.3) holds. Since

$$\phi'(x;d) = <\nabla f(x),d> + g'(x;d) \qquad \text{for all} \quad d\ ,$$

we have

$$g'(x;d) \geqq <-\nabla f(x),d> \qquad \text{for all} \quad d\ .$$

This implies by [R5 , Thm.23.1] that

$$[\ g(x+\lambda d) - g(x)\ ]\ /\ \lambda \geqq <-\nabla f(x),d> \qquad \text{for all} \quad \lambda > 0 \quad \text{and} \quad d\ ,$$

or equivalently,

$$g(x+\lambda d) \geqq g(x) + <-\nabla f(x),\lambda d> \qquad \text{for all} \quad \lambda > 0 \quad \text{and} \quad d\ .$$

This shows that $-\nabla f(x)$ is a subgradient of g at x.

The last assertion is immediate from the fact that (6.3) holds whenever x is local minimum of (6.1). □

If $\mathrm{dom}\, g = R^n$, then ϕ is locally Lipschitz, because it is the sum of an everywhere finite convex function and a continuously differentiable function, both of which are locally Lipschitz. By Clarke [C4], ϕ has a set of *generalized gradients*, which is denoted by $\partial^*\phi(x)$, at any point x . Moreover, it is not difficult to see that

$$\partial^*\phi(x) = \nabla f(x) + \partial g(x) .$$

Thus (6.3) is equivalent to the condition $0 \in \partial^*\phi(x)$, which is a necessary condition for x to be a local minimum of the locally Lipschitz function ϕ [C4]. (See also [P8].)

6.3 Algorithm

If ϕ is convex in problem (6.1), which is particularly true when f is convex, then it is known that a sequence $\{x_k\}$ generated by the following successive minimization algorithm converges to a minimum of ϕ : For each k , find the minimum, which is taken as x_{k+1} , of the function $\phi(x) + \frac{1}{2}c_k \| x - x_k \|^2$, where $c_k > 0$. Such an algorithm is called the *proximal point algorithm* or more specifically the *proximal minimization algorithm* and has recently been studied in detain by Rockafellar [R8] under a more general setting which allows inexact minimization at each iteration. For problem (6.1), however, the proximal point algorithm is not directly applicable since ϕ is in general nonconvex.

On the other hand, in the previous chapter, an algorithm of solving (6.1) has been proposed. It is briefly described as follows: For each k , find the minimum, which is denoted as $\tilde{x}_k$, of the *convex* function $\langle \nabla f(x_k), x \rangle + g(x)$ and then obtain x_{k+1} by the exact line search with respect ϕ along the direction $d_k = \tilde{x}_k - x_k$ from the current point x_k . It has been proved there that the sequence $\{x_k\}$ generated by the algorithm converges to a critical point of (6.1) under the strict convexity assumption on g .

The algorithm proposed here is a generalized version of the proximal point algorithm to cope with nonconvexity of the function ϕ by linearizing f at each iteration just as in the one proposed in the previous chapter. For each k , given x_k , the algorithm requires the solution of the subproblem

$$\text{minimize} \quad \langle \nabla f(x_k), x \rangle + \frac{c_k}{2} \| x - x_k \|^2 + g(x) \quad \text{over} \quad x \in R^n , \tag{6.4}$$

where c_k is a parameter such that $c_k > 0$.

Proposition 6.2. For any x_k and $c_k > 0$, problem (6.4) has the unique solution $\tilde{x}_k$ which can be expressed as

$$\tilde{x}_k = [I - c_k^{-1} \partial g]^{-1} [I - c_k^{-1} \nabla f] (x_k) .$$

Proof. Existence of the unique solution is obvious by the closedness and the strong convexity of the objective function of (6.4). Furthermore, by [R5 , Thms.23.5 and 23.8], $\tilde{x}_k$ is the minimum if and only if

$$0 \in \nabla f(x_k) + c_k (\tilde{x}_k - x_k) + \partial g(\tilde{x}_k) , \tag{6.5}$$

or equivalently

$$[I - c_k^{-1} \nabla f] (x_k) \in [I - c_k^{-1} \partial g] (\tilde{x}_k) .$$

Since $\tilde{x}_k$ is uniquely determined, the operator $[I - c_k^{-1} \partial g]^{-1}$ is single-valued, so that we obtain the required expression of $\tilde{x}_k$. □

Let α and β be scalar parameters such that $\alpha > 0$ and $0 < \beta < 1$, and let $\{c_k\}$ be a sequence of positive numbers. Given these data, the algorithm is formally stated as follows:

Algorithm.

Step 1: Choose an initial point $x_0 \in \text{dom } g$ and set $k = 0$.

Step 2: Find the solution x_k of (6.4) and set $d_k = \tilde{x}_k - x_k$.

Step 3: If $d_k = 0$, then terminate; otherwise, go to step 4.

Step 4: Compute $t_k = \beta^{\ell_k}$, where ℓ_k is the smallest nonnegative integer ℓ such that

$$\phi(x_k+\beta^{\ell}d_k) \leqq \phi(x_k) - \alpha\beta^{\ell}\| d_k\|^2 .$$

Step 5: Set $x_{k+1} = x_k + t_k d_k$, set $k = k+1$ and go to step 2.

Each step of the algorithm is explained as follows: Step 1 is the initialization. Step 2 generates a search direction at a current point by solving the subproblem. Step 3 is to check the optimality of the current point (see Propositions 6.3 and 6.4). Step 4 is to determine the step size t_k by the Armijo type rule [A5][P3]. Step 5 is the update of the solution.

It should be noted that the present algorithm is significantly different in principle from such general purpose algorithms as [F4] [L2][M4][W6] for minimizing nondifferentiable functions.

6.4 Convergence of Algorithm

In this section, we shall prove the convergence of the algorithm presented in the previous section. The next proposition validates the optimality test in step 3 of the algorithm.

<u>Proposition 6.3</u>. Let $\tilde{x}_k$ be the unique minimum of the subproblem (6.4), and let $d_k = \tilde{x}_k - x_k$. If $d_k = 0$, then x_k is a critical point of problem (6.1).

<u>Proof</u>. Suppose that $d_k = 0$, i.e., $\tilde{x}_k = x_k$. Then it follows from (6.5) that $0 \in \nabla f(x_k) + \partial g(x_k)$, which implies by definition that x_k is a critical point of (6.1). □

In order to simplify the notation, we define the operators P_k and Q_k from R^n into R^n by

$$P_k = [\ I + c_k^{-1}\partial g\ ]^{-1}$$

and

$$Q_k = I - c_k^{-1}\nabla f \ ,$$

respectively. It is known [R8 , p.878] that the operator P_k is nonexpansive, i.e.,

$$\| P_k(x) - P_k(y) \| \leq \| x - y \| \ , \quad \text{for all} \ x , y . \tag{6.6}$$

Moreover, let us define functions $\tilde{x}_k(\cdot)$ and $d_k(\cdot)$ by

$$\tilde{x}_k(x) = P_k Q_k(x)$$

and

$$d_k(x) = x_k(\tilde{x}) - x \ ,$$

respectively. It is not difficult to see that the functions $\tilde{x}_k(\cdot)$ and $d(\cdot)$ are continuous, since P_k is nonexpansive and Q_k is continuous. Moreover, they are Lipschitz continuous whenever ∇f is.

For any initial point x_0, let $L(x_0)$ denote the level set $\{x \in R^n;\ \phi(x) \leqq \phi(x_0)\}$. In the remainder of this chapter, we shall make use of the following assumptions.

<u>Assumption 6.1.</u> There exists some $L > 0$ such that for each $x_k \in L(x_0)$ and $d_k = d_k(x_k)$, we have

$$\| \nabla f(x_k+sd_k) - \nabla f(x_k) \| \leqq Ls \|d_k\| , \qquad 0 \leqq s \leqq 1 . \tag{6.7}$$

Clearly, assumption 6.1 is satisfied if ∇f is Lipschitz continuous on $L(x_0)$.

<u>Assumption 6.2.</u> For each $x_k \in L(x_0)$,

$$g'(x_k;d_k) = \liminf_{e \to d_k} g'(x_k;e) \tag{6.8}$$

and

$$g'(\tilde{x}_k;-d_k) = \liminf_{e \to d_k} g'(\tilde{x}_k;-e) , \tag{6.9}$$

where $\tilde{x}_k = \tilde{x}_k(x_k)$ and $d_k = d_k(x_k)$.

Since $g'(x;\cdot)$ is a positively homogeneous convex function for each fixed x [R5, Thm.23.1], (6.8) and (6.9) hold if $d_k(x_k) \in \text{ri dom } g'(x_k,\cdot)$ and $-d_k(x_k) \in \text{ri dom } g'(\tilde{x}_k;\cdot)$, respectively, or in particular $\text{dom } g'(x_k;\cdot) = \text{dom } g'(\tilde{x}_k;\cdot) = R^n$. Moreover, for any x_k and $\tilde{x}_k$ such that $x_k, \tilde{x}_k \in \text{ri dom } g$, (6.8) and (6.9) hold not only for $d_k = d_k(x_k)$ but also for all $d_k \in R^n$. Especially, if $\text{dom } g = R^n$, then assumption 6.2 is trivially satisfied.

Under assumption 6.2, it follows from [R2, p.503] that

$$g'(x_k;d_k) = \sup\{ \langle x^*,d_k\rangle \; ; \; x^* \in \partial g(x_k) \} \tag{6.10}$$

and

$$g'(\tilde{x}_k;-d_k) = \sup\{ \langle x^*,-d_k\rangle \; ; \; x^* \in \partial g(\tilde{x}_k) \} . \tag{6.11}$$

The following proposition shows that at any point x_k the direction d_k determined in step 2 of the algorithm is a direction of descent of the objective function ϕ.

<u>Proposition 6.4</u>. Let $x_k \in L(x_0)$ and $d_k = d_k(x_k)$. Then, under assumption 6.2, we have

$$\phi'(x_k;d_k) \leqq -c_k \|d_k\|^2 .$$

<u>Proof</u>. Throughout the proof, the subscript k is omitted for simplicity of notation. By (6.5), we have

$$-\nabla f(x) - cd \in \partial g(\tilde{x}) ,$$

where $\tilde{x} = \tilde{x}(x)$. By the monotonicity of the subdifferential operator ∂g [R5, Cor.31.5.2], we obtain

$$\langle -\nabla f(x)-cd-y,\tilde{x}-x\rangle \geqq 0 , \quad \text{for any } y \in \partial g(x) . \tag{6.12}$$

Since $d = \tilde{x} - x$, it follows from (6.10) and (6.12) that

$$\begin{aligned} \phi'(x;d) &= \langle \nabla f(x),d\rangle + g'(x;d) \\ &= \langle \nabla f(x),d\rangle + \sup\{ \langle y,d\rangle \; ; \; y \in \partial g(x) \} \\ &\leqq -c \| d \|^2 . \end{aligned}$$

□

Proposition 6.5. Let $x_k \in L(x_0)$ and $d_k = d_k(x_k)$. Suppose that $c_k > \alpha$. Then, under assumptions 6.1 and 6.2,

$$\phi(x_k+td_k) \leqq \phi(x_k) - \alpha t \|d_k\|^2$$

holds for all t such that $0 \leqq t \leqq \min \{ 1, \frac{2(c_k-\alpha)}{L} \}$.

Proof. Throughout the proof, the subscript k is omitted for simplicity of notation. By (6.7), we have

$$\begin{aligned} f(x+td) - f(x) &= \int_0^t \langle \nabla f(x+sd), d\rangle \, ds \\ &= t \langle \nabla f(x), d\rangle + \int_0^t \langle \nabla f(x+sd) - \nabla f(x), d\rangle \, ds \\ &\leqq t \langle \nabla f(x), d\rangle + \int_0^t Ls \|d\|^2 \, ds \\ &= t \langle \nabla f(x), d\rangle + \frac{1}{2} Lt^2 \|d\|^2 . \end{aligned}$$

Let us define the function $\theta: [0.1] \to R$ by $\theta(s) = g(x+sd)$. It is noted that $\theta(s) < +\infty$ since both x and $\tilde{x}$ are in dom g. Let $\theta'_+(s)$ and $\theta'_-(s)$ denote the right and left derivetive functions [R5, p.214], respectively. Then

$$\theta'_+(s) = g'(x+sd;d) \quad \text{and} \quad \theta'_-(s) = -g'(x+sd;-d) ,$$

and by [R5, Thm.24.1],

$$\theta'_+(s_1) \leqq \theta'_-(s) \leqq \theta'_+(s) \leqq \theta'_-(s_2) \qquad \text{when} \quad s_1 < s < s_2 .$$

In particular, we have

$$g'(x+sd;d) \leqq -g'(\tilde{x};-d) \qquad \text{for all} \quad 0 \leqq s < 1 ,$$

since $\tilde{x} = x + d$. Thus it follows from [R5 , Cor.24.2.1] that for any $0 \leqq t < 1$

$$g(x+td) - g(x) = \int_0^t g'(x+sd;d)\, ds$$

$$\leqq - t\, g'(\tilde{x};-d) \ .$$

Since g is lower semicontinuous, it is not difficult to see that the last relation is also valid for $t = 1$. On the other hand, by (6.11), we have

$$- g'(\tilde{x};-d) = \inf \{ <x^*,d> \ ; \ x^* \in \partial g(\tilde{x}) \}$$

$$\leqq - c \|d\|^2 - <\nabla f(x),d> \ ,$$

since $- cd - \nabla f(x) \in \partial g(\tilde{x})$, so that

$$g(x+td) - g(x) \leqq - ct\|d\|^2 - t <\nabla f(x),d> \ .$$

Consequently, for $0 \leqq t \leqq 1$

$$\phi(x+td) - \phi(x) = [f(x+td) - f(x)] + [g(x+td) - g(x)]$$

$$\leqq (- ct + \frac{Lt^2}{2}) \|d\|^2 \ .$$

Hence the result follows immediately (see Fig.6.1). □

The following corollary ensures that the step size t_k chosen in step 4 by the rule of Armijo type is bounded away from zero. The proof is omitted since the result is an immediate consequence of Proposition 6.5 and the construction of t_k in step 4.

<u>Corollary</u>. Let all the assumptions in Proposition 6.5 be satis-

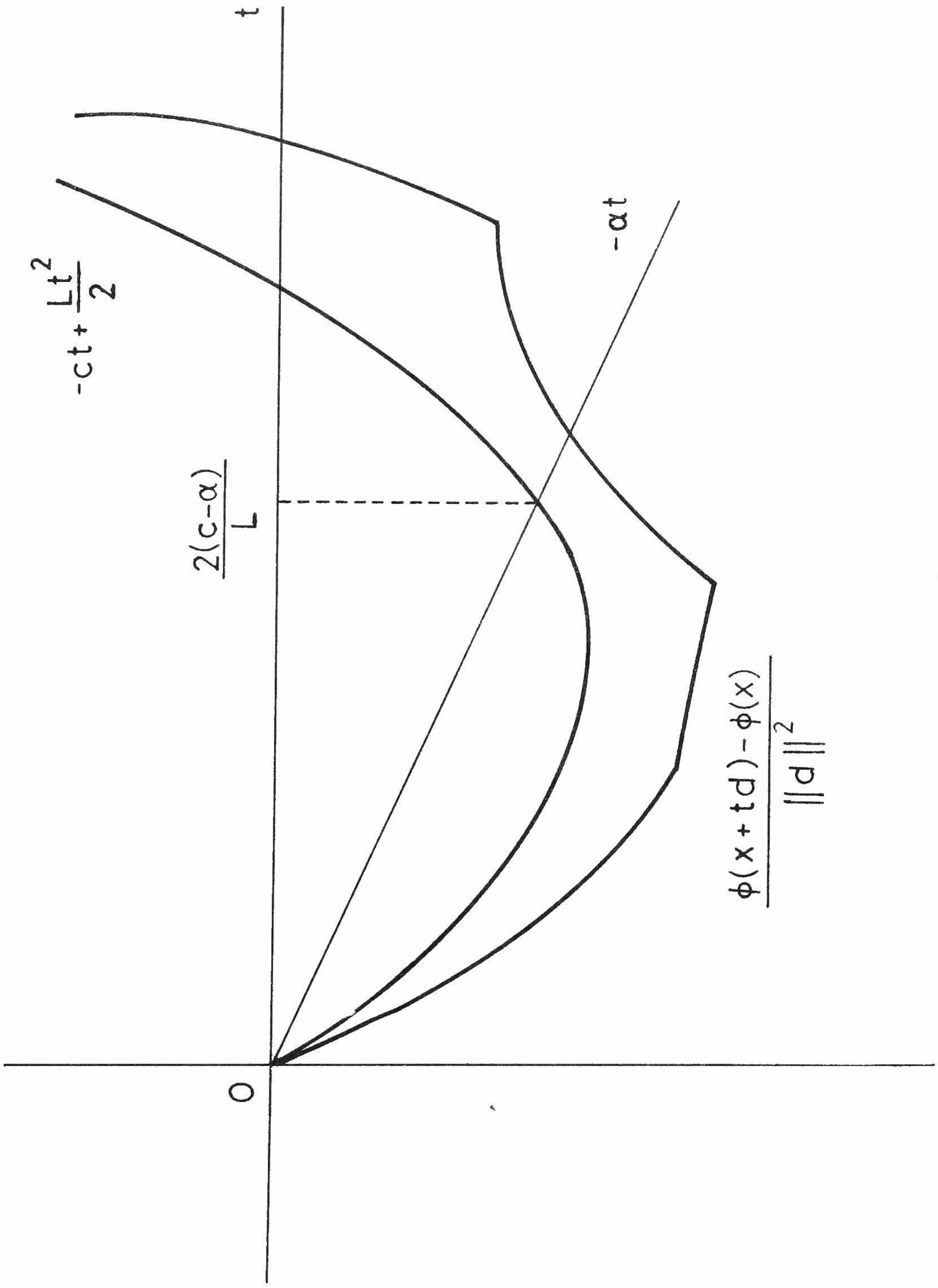


FIG. 6.1

fied. Suppose that there exists some c such that $c_k \geqq c > \alpha$ for all k. Then, for each k, the step size t_k determined by the algorithm satisfies

$$t_k \geqq \min\{1, \frac{2(c-\alpha)}{L}\} . \tag{6.13}$$

In particular, if $c - \alpha \geqq \frac{\beta L}{2}$, then $x_{k+1} = \tilde{x}_k$ for all k.

Proposition 6.6. In addition to all the assumptions in the corollary of Proposition 6.5, assume that ϕ is bounded below. Then as $k \to \infty$, we have

$$\| d_k \| \longrightarrow 0$$

and

$$\| x_{k+1} - x_k \| \longrightarrow 0 .$$

Proof. Since the sequence $\{\phi(x_k)\}$ is monotonically decreasing and is bounded from below, we have

$$\phi(x_{k+1}) - \phi(x_k) \longrightarrow 0 \qquad \text{as} \quad k \to \infty .$$

Thus by (6.13), we obtain

$$\begin{aligned} \|d_k\|^2 &\leqq \{ \phi(x_{k+1}) - \phi(x_k) \} / \alpha t_k \\ &\leqq \{ \phi(x_{k+1}) - \phi(x_k) \} / \alpha \min\{1, \frac{2(c-\alpha)}{\beta L}\} \\ &\longrightarrow 0 . \end{aligned}$$

Convergence of the sequence $\{ \|x_{k+1} - x_k\| \}$ to zero is obvious since

$$\| x_{k+1} - x_k \| = t_k \|d_k\| \leqq \|d_k\| . \qquad \square$$

Now we state the main result of this section.

Theorem 6.1. Let x_0 be any initial point. Assume that ϕ is bounded from below and that the sequence $\{c_k\}$ is bounded above and away from α, i.e., there exist some $\bar{c}$ and c such that $\bar{c} \geqq c_k \geqq c > \alpha$ for all k. Suppose that assumptions 6.1 and 6.2 are satisfied. Then any accumulation point of the sequence $\{x_k\}$ generated by the algorithm is a critical point of problem (6.1).

Proof. Let $\bar{x}$ be any accumulation point of $\{x_k\}$ and let $\{x_{k_i}\}$ be a subsequence of $\{x_k\}$ converging to $\bar{x}$. First note, by Proposition 6.6, that

$$\| d_{k_i} \| = \| \tilde{x}_{k_i} - x_{k_i} \| \longrightarrow 0$$

which clearly implies

$$\tilde{x}_{k_i} \longrightarrow \bar{x} \quad . \tag{6.14}$$

Since ∇f is continuous and $\{c_{k_i}\}$ is bounded,

$$- \nabla f(x_{k_i}) - c_{k_i}(\tilde{x}_{k_i} - x_{k_i}) \longrightarrow - \nabla f(\bar{x}) \quad . \tag{6.15}$$

By (6.5), we have

$$- \nabla f(x_{k_i}) - c_{k_i}(\tilde{x}_{k_i} - x_{k_i}) \in \partial g(x_{k_i}) \quad . \tag{6.16}$$

It then follows from (6.14), (6.15) and (6.16) that

$$- f(\bar{x}) \in \partial g(\bar{x}) \quad ,$$

by virtue of the upper semicontinuity of the point-to-set map ∂g [R5, Thm.24.4]. This completes the proof. □

Note that Theorem 6.1 says nothing about the existence of accumulation points of $\{x_k\}$. In fact, Theorem 6.1 states that if there *exists* a convergent subsequence of $\{x_k\}$, then its limit must be a critical point. In order to assure the existence of the limit, we need some additional assumption such as the compactness of the level set $L(x_0)$.

Corollary. Let all the assumptions in Theorem 6.1 be satisfied. Suppose, in addition, that there exist a finite number of critical points and that the sequence $\{x_k\}$ is bounded. Then $\{x_k\}$ converges to one of the critical points.

Proof. Because of the boundedness of $\{x_k\}$, the existence of at least one accumulation point is assured. Note by Proposition 6.6 that

$$\| x_{k+1} - x_k \| \longrightarrow 0 \quad .$$

Then the corollary may be proved in a manner similar to [Ol , 14.1.5]. □

6.5 Rate of Convergence

Throughout this section, we assume that the sequence $\{x_k\}$ generated by the algorithm has a unique accumulation point $\bar{x}$, i.e., $x_k \to \bar{x}$. Recall that the sequence $\{x_k\}$ is said to *converge linearly to* $\bar{x}$ *with ratio* γ if

$$\limsup_{k \to \infty} \frac{\| x_{k+1} - \bar{x} \|}{\| x_k - \bar{x} \|} = \gamma < 1 .$$

The purpose of this section is to show that $\{x_k\}$ converges linearly to $\bar{x}$ under certain conditions. We have proved in Section 6.4 the convergence of the algorithm in which the sequence $\{c_k\}$ was assumed to be bounded. In the convergence rate analysis of this section, we shall restrict our attention to a special case in which $c_k \equiv c$.

For any $n \times n$ matrix A, a norm of A may be defined by

$$\| A \| = \sup_{\| x \| = 1} \| Ax \| .$$

If A is symmetric with eigenvalues $\lambda_1, \ldots, \lambda_n$, then we have ([Ol, p.41])

$$\| A \| = \max \{ |\lambda_1|, \ldots, |\lambda_n| \} .$$

Lemma 6.1. Suppose that A is symmetric and positive definite. Then

$$\| I - c^{-1}A \| < 1$$

holds provided that $c > \frac{1}{2} \|A\|$.

Proof. Assume without loss of generality that

$$0 < \lambda_1 \leqq \ldots \leqq \lambda_n = \| A \| ,$$

where λ_i are the eigenvalues of A. Note that $I - c^{-1}A$ is symmetric with eigenvalues $1 - c^{-1}\lambda_i$, $i=1,\ldots, n$. Thus if $c > \frac{1}{2}\|A\|$, then we obtain

$$-1 < 1 - c^{-1}\lambda_n \leqq \ldots \leqq 1 - c^{-1}\lambda_1 < 1 ,$$

which implies $\|I - c^{-1}A\| < 1$. □

Assumption 6.3. f is twice differentiable at $\bar{x}$ and the Hessian matrix $\nabla^2 f(\bar{x})$ is positive definite.

Theorem 6.2. Let $\{x_k\}$ be a sequence generated by the algorithm in which $c_k \equiv c$. Suppose that $\{x_k\}$ converges to $\bar{x}$ and that assumptions 6.1, 6.2 and 6.3 are satisfied. If

$$c > \max \{ \alpha , \frac{1}{2} \|\nabla^2 f(\bar{x}) \| \} ,$$

then x_k converges linearly with ratio

$$\gamma \leqq \max \{ \sigma , 1 - \frac{2(c - \alpha)(1 - \sigma)}{\beta L} \} < 1 ,$$

where $\sigma = \| I - c^{-1}\nabla^2 f(\bar{x}) \|$.

Proof. First note that $P_k \equiv P$ and $Q_k \equiv Q$, where

$$P = [I - c^{-1}\partial g]^{-1}$$

and

$$Q = I - c^{-1}\nabla f .$$

Then by definition, $\tilde{x}_k = PQ(x_k)$. Moreover, since the accumulation point $\bar{x}$ satisfies (6.2), it satisfies the fixed point property $\bar{x} = PQ(\bar{x})$. Thus by the nonexpansiveness (6.6) of P , we obtain

$$\begin{aligned} \| \tilde{x}_k - \bar{x} \| &= \| PQ(x_k) - PQ(\bar{x}) \| \\ &\leqq \| Q(x_k) - Q(\bar{x}) \| \\ &\leqq \| Q(x_k) - Q(\bar{x}) + \nabla Q(\bar{x})(x_k - \bar{x}) \| \\ &\quad + \| \nabla Q(\bar{x})(x_k - \bar{x}) \| , \end{aligned}$$

where $\nabla Q(\bar{x}) = I - c^{-1}\nabla^2 f(\bar{x})$ which exists by assumption 6.3. Let ε be an arbitrary positive number. Since $\{x_k\} \rightarrow \bar{x}$, we have for all k sufficiently large

$$\| Q(x_k) - Q(\bar{x}) - \nabla Q(\bar{x})(x_k - \bar{x}) \| \leqq \varepsilon \| x_k - \bar{x} \| .$$

On the other hand, by the definition of the matrix norm,

$$\| \nabla Q(\bar{x})(x_k - \bar{x}) \| \leqq \sigma \| x_k - \bar{x} \| ,$$

where $\sigma = \| \nabla Q(\bar{x}) \|$. Hence we obtain

$$\| \tilde{x}_k - \bar{x} \| \leqq (\sigma + \varepsilon) \| x_k - \bar{x} \| ,$$

from which it follows that

$$\begin{aligned} \| x_{k+1} - \bar{x} \| &= \| x_k + t_k d_k - \bar{x} \| \\ &= \| t_k \tilde{x}_k + (1 - t_k) x_k - \bar{x} \| \\ &\leqq t_k \| \tilde{x}_k - \bar{x} \| + (1 - t_k) \| x_k - \bar{x} \| \\ &\leqq [1 - t_k (1 - \sigma - \varepsilon)] \| x_k - \bar{x} \| \qquad (6.17) \end{aligned}$$

for all k sufficiently large. Thus if $c > \alpha$, then by (6,13) and (6.17),

$$\| x_{k+1} - \bar{x} \| \leqq \gamma_\varepsilon \| x_k - \bar{x} \|$$

holds for sufficiently large k, where

$$\gamma_\varepsilon = \max \{ \sigma + \varepsilon , 1 - \frac{2(c-\alpha)(1-\sigma-\varepsilon)}{\beta L} \}$$

Since ε is arbitrary, we obtain asymptotically

$$\gamma = \limsup_{k \to \infty} \frac{\| x_{k+1} - \bar{x} \|}{\| x_k - \bar{x} \|} \leqq \gamma_0 ,$$

where

$$\gamma_0 = \max \{ \sigma , 1 - \frac{2(c-\alpha)(1-\sigma)}{\beta L} \} . \tag{6.18}$$

Moreover, if $c > \frac{1}{2} \|\nabla^2 f(\bar{x})\|$, then Lemma 6.1 assures that $\sigma < 1$, which in turn imlies that right hand side of (6.18) is also less than one. This completes the proof. □

6.6 Discussions

If $f \equiv 0$ in problem (6.1), the proposed algorithm relates very closely to the proximal point algorithm. In fact, for such cases, it is shown in view of Proposition 6.5 and its corollary, that $t_k \equiv 1$ provided that $\{c_k\}$ is chosen so that $c_k > \alpha$ (note that we may suppose $L = 0$). Hence, the proposed algorithm coincides with the proximal point algorithm with exact minimization. Rockafellar [R8] shows that the rate of convergence can be made superlinear by letting $c_k \longrightarrow 0$ (note that here c_k is the inverse of that in [R8]). For general cases, i.e., $f \not\equiv 0$, however, the rate of convergence of the proposed algorithm cannot be improved by controlling $\{c_k\}$, because it is impossible to let $c_k \longrightarrow 0$ due to the feasibility of the Armijo type step size rule.

On the other hand, if $g \equiv 0$ in problem (6.1), the situation becomes considerably simpler. Indeed, by (6.5), we have

$$d_k = \tilde{x}_k - x_k = -c_k^{-1}\nabla f(x_k) ,$$

which shows that d_k is the direction of steepest descent. Hence, the proposed algorithm coincides with the steepest descent method with the Armijo step size rule [A5] [P3].

Finally let us compare the proposed algorithm with 'subgradient' methods. For problem (6.1), a natural extension of subgradient methods, e.g., [P4] [P5], of minimizing, possibly nondifferentiable, convex functions might be to choose a search direction d_k at a current point x_k among the set $-\nabla f(x_k) - \partial g(x_k)$. It should be noted that the search direction so chosen is not in general a direction of descent of ϕ , so that a line search such as that of Armijo type may be infeasible. On the other hand, the proposed algorithm determines d_k such that

$$c_k d_k = c_k(\tilde{x}_k - x_k) \in -\nabla f(x_k) - \partial g(\tilde{x}_k) ,$$

which is, by Proposition 6.4, a direction of descent of ϕ .

6.7 Conclusion

We have proposed the generalized proximal point algorithm for a certain type of nonconvex minimization problems. The algorithm presented in the previous chapter and this algorithm have several characteristics in common. However, they are significantly different in three aspects, i.e., (i) the latter exploits the finite procedure of Armijo type for the line search, while the former requires in general an infinite number of steps for the exact line search, (ii) the convergence of the former has been proved under the strict convexity assumption on the function involved, while the latter does not require such strictness, (iii) each auxiliary problem solved in the latter always contains a quadratic term, while it is not neccessarily the case for the former. Consequently, it is advised that we may use either the present algorithm or that proposed in Chapter 5 according to the structure of the problem to solve.

CHAPTER 7

MULTILEVEL DECOMPOSITION OF NONLINEAR PROGRAMMING PROBLEMS BY DYNAMIC PROGRAMMING

In this chapter, we are concerned with large-scale nonlinear programming problems. In particular, we study in detail a multilevel structure in nonlinear programming problems that can be solved by dynamic programming, and derive recursive functional equations that combine optimal solutions of subproblems to yield an optimal solution of the principal problem. The analysis developed in this chapter is very general and will be helpful in the discussion of Chapters 8 and 9.

7.1 Introduction

In various fields of engineering and economics, it is often necessary to deal with large-scale mathematical programming problems. Up to the present, many sorts of decomposition techniques [D2][G4][H3] [Ll][M3][S3] have been contrived to solve such problems. Especially, dynamic programming originated by Bellman [B2][B3] is one of the effective techniques for complex optimization problems. However, in order that the problem be solved by dynamic programming, we must break it down into a series of subproblems and then combine the optimal solutions of the subproblems to find the optimal solution of the original problem. The first part of this procedure is decomposition of the problem and the second part is composition by functional equations that reveal the principle of optimality of dynamic programming.

The purpose of this chapter is to clarify the applicability of dynamic programming to general nonlinear programming problems. It is noted that Mitten [M16] and Nemhauser [N1] established some sufficient conditions for multistage decision problems to be decomposed by dynamic programming by making use of the notion of composition operators. On the other hand, Mine and Ohno [M13] show that mathematical programming problems may be considered as multistage decision problems under certain conditions, and give recursive functional equations of dynamic programming for these problems. Here, we consider a multilevel structure of mathematical programming problems that can be solved by dynamic programming and present recursive functional equations that combine optimal solutions of subproblems to obtain an optimal solution of the original problem. The sufficient condition for the decomposition by dynamic programming given in this chapter seems to be the weakest one, as far as mathematical programming problems are concerned.

This chapter is organized as follows: In Section 7.2, we give some preliminary definitions which are fundamental in the subsequent discussions. In Section 7.3, we show the main results on decomposition of the problem by dynamic programming. In Section 7.4, we solve a quadratic fractional programming problem as a numerical example. In Section 7.5, we indicate that dynamic programming is frequently applicable to the nonlinear programming problem if the functions involved are continuous.

7.2 Decomposability and Separability

In this section, a decomposability and a separability of functions are introduced, which are fundamental concepts required throughout this chapter.

To begin with, the decomposability and the disjoint decomposability of functions are defined. It is shown that the decomposable functions can be reduced to the disjointly decomposable ones by some modifications. Moreover, the sequential separability and the monotonicity of functions are defined.

Definition 7.1. Let $H(x_1,\dots,x_N)$ and $H_{ij}(z_1,\dots,z_{|I_{ij}|})$, $i=1,\dots,s$, $j=1,\dots,n_i$, be vector valued functions of vectors x_n, $n=1,\dots,N$, and of vectors z_k, $k=1,\dots,|I_{ij}|$, respectively, where $n_1 = N$, $n_s = 1$, and I_{ij} are sets of indices such that $I_{1j} = \{j\}$, $j=1,\dots,n_1$, and for $i = 2,\dots,s$, $I_{ij} \subset \{1,\dots,n_{i-1}\}$ and $\cup_{i=1}^{n_i} I_{ij} = \{1,\dots,n_{i-1}\}$; $|I_{ij}|$ denotes the number of elements of I_{ij}. $H(x_1,\dots,x_N)$ is said to be *(s-level-)decomposable* by the family of functions $\{H_{ij};\ i=1,\dots,s,\ j=1,\dots,n_i\}$, if recursive relations

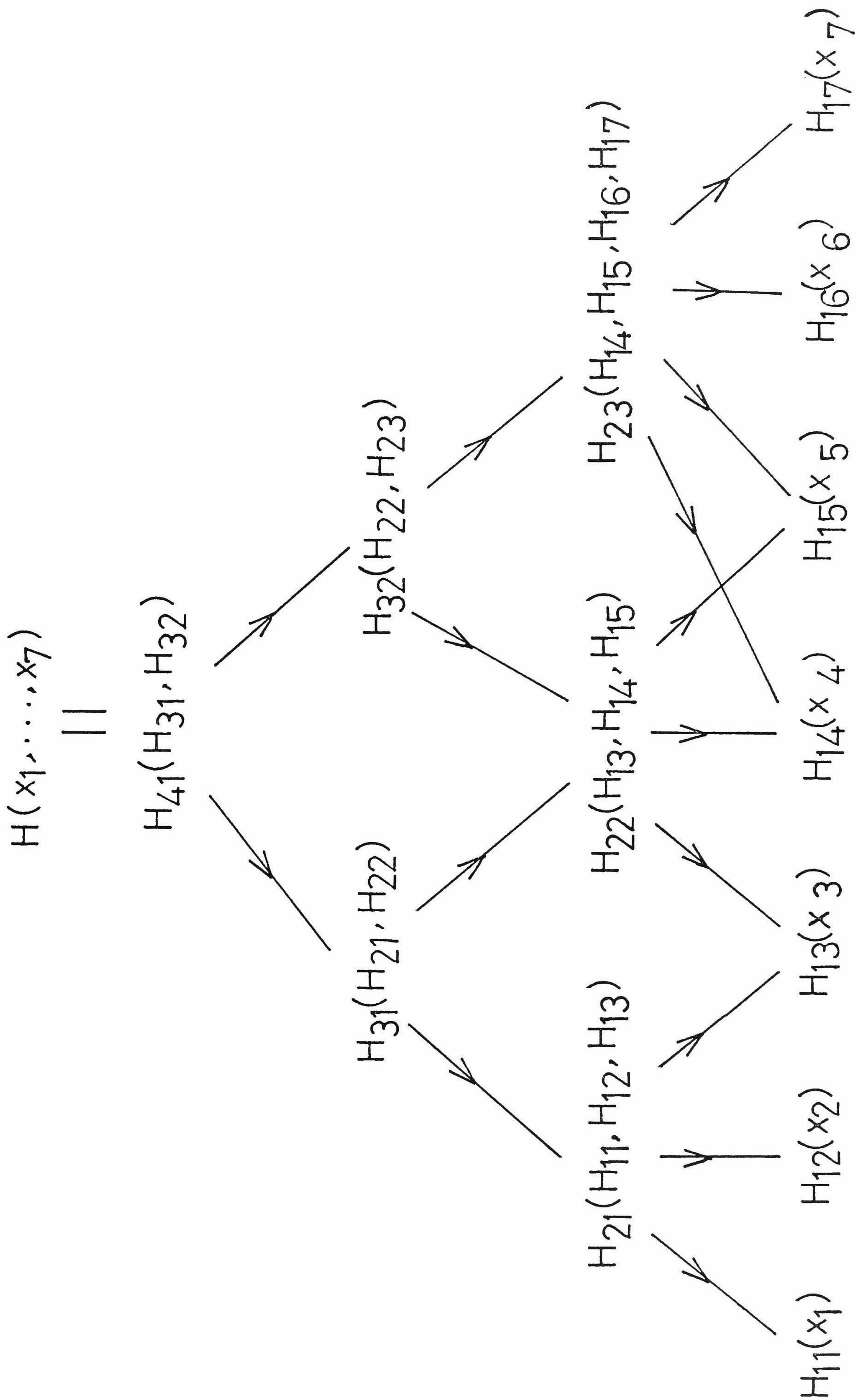
$$H_{1j} = H_{1j}(x_j), \qquad j=1,\dots,n_1,$$

$$H_{ij} = H_{ij}(A_{ij}), \qquad i=1,\dots,s,\ j=1,\dots,n_i,$$

and

$$H_{s1} = H(x_1,\dots,x_N)$$

hold, where for $I_{ij} = \{j_1,\dots,j_{|I_{ij}|}\}$, $A_{ij} = \{H_{i-1j_1},\dots, H_{i-1j_{|I_{ij}|}}\}$ and $H_{ij}(A_{ij})$ means $H_{ij}(H_{i-1j_1},\dots, H_{i-1j_{|I_{ij}|}})$.


H(x1, . . . , x7)
H41(H31, H32)
H31(H21, H22)
H32(H22, H23)
H21(H11, H12, H13)
H22(H13, H14, H15)
H23(H14, H15, H16, H17)
H11(x1)
H12(x2)
H13(x 3)
H14(x 4)
H15(x 5)
H16(x 6)
H17(x 7)

In general, the decomposition may not be unique. To see the decomposition scheme, it is useful to represent it by a graph. This graph is determined as a directed graph consisting of the vertices $\{H_{ij}\ ;\ i=1,\dots,s,\ j=1,\dots,n_i\}$ and the edges $\{(H_{ij}, H_{i-1k})\ ;\ H_{i-1k} \in A_{ij}\ ,\ i=2,\dots,s\ ,\ j=1,\dots,n_i\}$. An example of the graph is shown in Fig. 7.1. In this example, $n_4 = 1$, $I_{41} = \{1,2\}$; $n_3 = 2$, $I_{31} = \{1,2\}$, $I_{32} = \{2,3\}$; $n_3 = 3$, $I_{21} = \{1,2,3\}$, $I_{22} = \{3,4,5\}$, $I_{23} = \{4,5,6,7\}$; $n_1 = 7$, $I_{11} = \{1\}$, $I_{12} = \{2\}$, $I_{13} = \{3\}$, $I_{14} = \{4\}$, $I_{15} = \{5\}$, $I_{16} = \{6\}$, $I_{17} = \{7\}$.

Definition 7.2. If the sets A_{ij} , $i=2,\dots,s$, $j=1,\dots,n_i$, in Definition 7.1 satisfy the property that $j \neq j'$ implies $A_{ij} \cap A_{ij'} = \emptyset$, then the decomposition is said to be *disjoint*.

A similar notion has been used by Karp [K1] in connection with switching functions. A graph corresponding to a disjoint decomposition is a tree. An example is shown in Fig. 7.2.

Now let us note that an arbitrary decomposable function can be reduced to a disjointly decomposable one by adding appropriate variables to the original function. Let the number of different paths from the vertex H_{s1} to H_{ij} be σ_{ij} . For example, in Fig. 7.1, $\sigma_{11} = 1$, $\sigma_{12} = 1$, $\sigma_{13} = 3$, $\sigma_{14} = 3$, $\sigma_{15} = 3$, $\sigma_{16} = 1$, $\sigma_{17} = 1$, $\sigma_{21} = 1$, $\sigma_{22} = 2$, $\sigma_{23} = 1$, $\sigma_{31} = 1$, $\sigma_{32} = 1$. Put additional variables $x_j^{(p)}$, $p=1,\dots,\sigma_{ij}$, and functions $H_{ij}^{(p)}$, $p=1,\dots,\sigma_{ij}$, as follows:

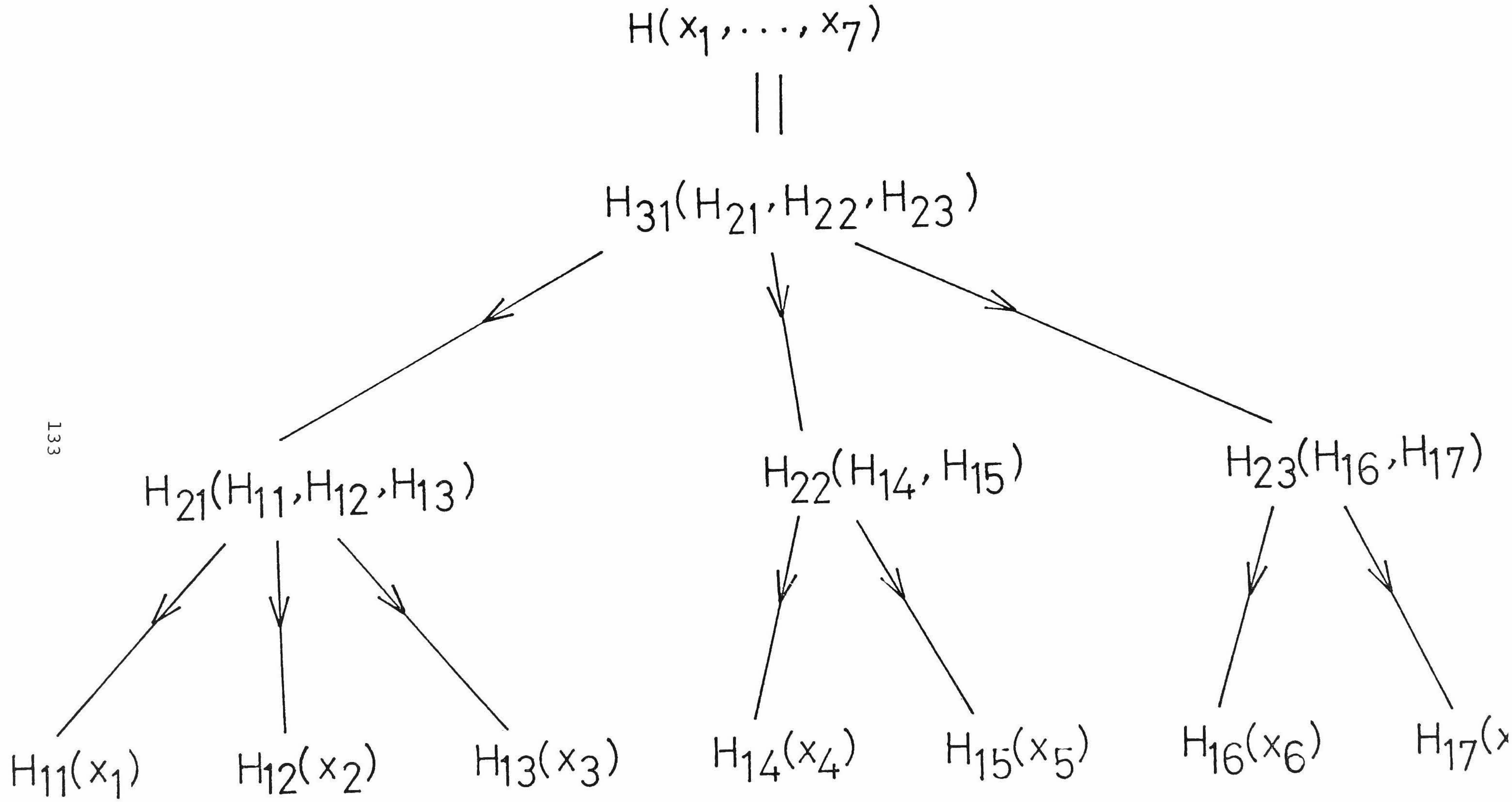


Fig. 7.2. Disjoint decomposition

$$x_j = x_j^{(1)} = \ldots = x_j^{(\sigma_{1j})} , \qquad j=1,\ldots,n_1 ,$$

and

$$H_{ij} = H_{ij}^{(1)} = \ldots = H_{ij}^{(\sigma_{ij})} , \qquad i=1,\ldots,s , \; j=1,\ldots,n_i .$$

Then redefine the decomposition of function H in the following manner:

$$H_{1j}^{(p)} = H_{1j}^{(p)}(x_j^{(p)}) , \qquad j=1,\ldots,n_i , \; p=1,\ldots,\sigma_{1j} ,$$

$$H_{ij}^{(p)} = H_{ij}^{(p)}(A_{ij}^{(p)}) , \qquad i=2,\ldots,s , \; j=1,\ldots,n_i , \; p=1,\ldots,\sigma_{ij} ,$$

and

$$H_{s1}^{(1)} = H(x_1,\ldots,x_N) ,$$

where for $p = 1,\ldots, \sigma_{ij}$,

$$A_{ij}^{(p)} = \cup_{k\in I_{ij}} \{H_{i-1k}^{(p)} , \; 1 \leqq q \leqq \sigma_{i-1k}\} ,$$

and furthermore $H_{i-1k}^{(q)} \in A_{ij}^{(p)}$ and $H_{i-1k}^{(q')} \in A_{ij'}^{(p')}$ imply that if $j < j'$ or $j = j'$ and $p < p'$, then $q < q'$.

Since $j \neq j'$ or $p \neq p'$ clearly implies $A_{ij}^{(p)} \cap A_{ij'}^{(p')} = \emptyset$, this modified decomposition is disjoint. For example, the graph shown in Fig. 7.1 can be transformed into the graph corresponding to a disjoint decomposition shown in Fig. 7.3.

Definition 7.3. Let $H(y_1,\ldots,y_k)$ be a vector valued function of n_k-vectors y_k , $k=1,\ldots,K$. The function H is said to be *sequentially separable,* if there exists a sequence of vector valued functions $\{h^1,\ldots,h^K\}$ such that vector valued functions $H^k(y_1,\ldots,y_k)$

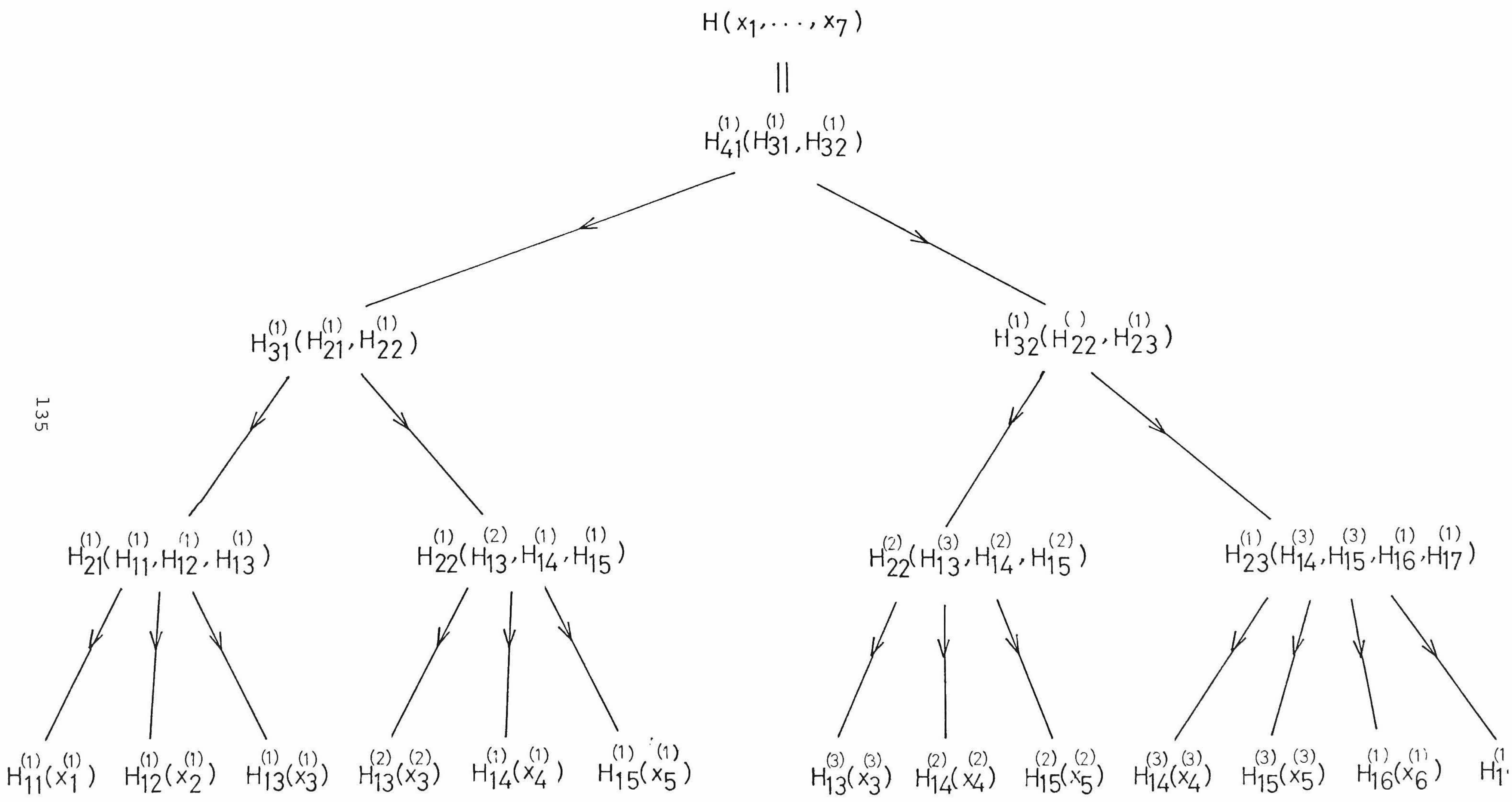


Fig. 7.3. Reduction to disjoint decomposition

defined recursively by

$$H^1(y_1) = h^1(y_1)$$

and

$$H^k(y_1,\ldots,y_k) = h^k(H^{k-1}(y_1,\ldots,y_{k-1}),y_k) \ , \qquad k=2,\ldots,K \ ,$$

satisfy

$$H^K(y_1,\ldots,y_K) = H(y_1,\ldots,y_K) \ .$$

Moreover, the functions h^k are called *the separating functions* of H .

Definition 7.4. Let y_k , $k=1,\ldots,K$, be real numbers and $H(y_1,\ldots,y_K)$ be a real valued function defined on a subset of R^K . Let H be sequentially separable with real valued separating functions $\{h^k(H^{k-1},y_k) \ ; k=2,\ldots,K\}$. If each $h^k(H^{k-1},y_k)$ is a non-decreasing function with respect to both H^{k-1} and y_k , then the sequential separation of $H(y_1,\ldots,y_K)$ is said to be *monotone.*

7.3 Main Results

The problem to be considered is

$$\begin{aligned} &\text{maximize} && F(x_1,\dots,x_N) \\ &\text{subject to} && G(x_1,\dots,x_N) = 0 \\ &\text{and} && x_n \in X_n , \quad n=1,\dots,N , \end{aligned} \tag{7.1}$$

where X_n , $n=1,\dots,N$, is a subset of R^{k_n} , x_n , $n=1,\dots,N$, is a k_n-vector, the objective function $F(x_1,\dots,x_N)$ is a real valued function defined on $\times_{i=1}^{N} X_n$, and the constraint function $G(x_1,\dots,x_N)$ is an M-dimensional vector valued function $(G_1(x_1,\dots,x_N),\dots, G_M(x_1,\dots,x_N))$ defined on $\times_{i=1}^{N} X_n$. In what follows, this problem is referred to as *the principal problem* and is abbreviated to

$$P \equiv \max\{\, F(x_1,\dots,x_N) \mid G(x_1,\dots,x_N) = 0 \text{ and } x_n \in X_n ,\ n=1,\dots,N \,\} .$$

The purpose of this section is to present sufficient conditions for the principal problem to be decomposed into subproblems by dynamic programming. For the principal problem (7.1), let the objective function $F(x_1,\dots,x_N)$ and the constraint function $G(x_1,\dots,x_N)$ be decomposable by the family $\{F_{ij}\}$ and by the family $\{G_{ij}\}$, respectively, as follows:

$$\begin{aligned} &F_{1j} = F_{1j}(x_j) , && j=1,\dots,N , \\ &F_{ij} = F_{ij}(A_{ij}) , && i=2,\dots,s ,\ j=1,\dots,n_i , \\ &F_{s1} = F(x_1,\dots,x_N) , && \end{aligned}$$

and

$$G_{1j} = G_{1j}(x_j) , \qquad j=1,\dots,N ,$$

$$G_{ij} = G_{ij}(B_{ij}) , \qquad i=2,\dots,s', \ j=1,\dots,n_i ,$$

$$G_{s'1} = G(x_1,\dots,x_N) ,$$

where $A_{ij} = \cup_{k\in I_{ij}}\{F_{i-1k}\}$ and $B_{ij} = \cup_{k\in J_{ij}}\{G_{i-1k}\}$. The principal problem is said to be *decomposable* if there exist decompositions of F and G such that $s = s'$ and for all i and j , $I_{ij} = J_{ij}$, that is, if F and G have the similar structure. The *disjoint decomposability* of the principal problem is also defined in a manner similar to Definition 7.2. In decomposed problems, F_{ij} and G_{ij} , $i=1,\dots,s$, $j=1,\dots,n_i$, become the objective function and the constraint function of the jth subproblem in the ith level, respectively.

Theorem 7.1. For the disjointly decomposable problem, assume that F_{ij} and G_{ij} , $i=2,\dots,s$, $i=1,\dots,n_i$, are sequentially separable and that the separations of F_{ij} are monotone. Then the principal problem is decomposed into subproblems by dynamic programming.

Proof. The objective and the constraint functions of the jth subproblem in the ith level ($i=2,\dots,s$, $j=1,\dots,n_i$) are written as

$$F_{ij} = F_{ij}(F_{i-1j_1},\dots,F_{i-1j_{|I_{ij}|}})$$

and

$$G_{ij} = G_{ij}(G_{i-1j_1},\dots,G_{i-1j_{|I_{ij}|}}) ,$$

respectively. From the assumption of the sequential separability, there exist separating functions f_{ij}^{n} and g_{ij}^{n}, $n=1,\dots,|I_{ij}|$, such that

$$F_{ij}^{1}(F_{i-1j_1}) = f_{ij}^{1}(F_{i-1j_1}) ,$$

$$F_{ij}^{n}(F_{i-1j_1},\dots,F_{i-1j_n}) = f_{ij}^{n}(F_{ij}^{n-1}(F_{i-1j_1},\dots,F_{i-1j_{n-1}}),F_{i-1j_n}) ,$$

$$n = 2,\dots,|I_{ij}| ,$$

$$F_{ij}^{|I_{ij}|}(F_{i-1j_1},\dots,F_{i-1j_{|I_{ij}|}}) = F_{ij}(F_{i-1j_1},\dots,F_{i-1j_{|I_{ij}|}}) ,$$

and

$$G_{ij}^{1}(G_{i-1j_1}) = g_{ij}^{1}(G_{i-1j_1}) ,$$

$$G_{ij}^{n}(G_{i-1j_1},\dots,G_{i-1j_n}) = g_{ij}^{n}(G_{ij}^{n-1}(G_{i-1j_1},\dots,G_{i-1j_{n-1}}),G_{i-1j_n}) ,$$

$$n = 2,\dots,|I_{ij}| ,$$

$$G_{ij}^{|I_{ij}|}(G_{i-1j_1},\dots,G_{i-1j_{|I_{ij}|}}) = G_{ij}(G_{i-1j_1},\dots,G_{i-1j_{|I_{ij}|}}) .$$

Put for $n = 1,\dots,|I_{ij}|$,

$$P_{ij}^{n}(z_n) \equiv \max\{ F_{ij}^{n}(F_{i-1j_1},\dots,F_{i-1j_n}) \mid G_{ij}^{n}(G_{i-1j_1},\dots,G_{i-1j_n}) = z_n \} , \tag{7.2}$$

$$P_{ij}(z) \equiv P_{ij}^{|I_{ij}|}(z) = \max\{ F_{ij} \mid G_{ij} = z \} , \tag{7.3}$$

$$S_{ij}^{n}(z_n) \equiv \{(u_{i-1j_1},\dots,u_{i-1j_n}) ; G_{ij}^{n}(u_{i-1j_1},\dots,u_{i-1j_n}) = z_n\} ,$$

$$S_{ij}(z) \equiv \{ u_{ij} ; u_{ij} = z \} ,$$

$$V^n_{ij}(z_n) \equiv \{ \, y_n \; ; \text{ there exists } \; z_{n-1} \; \text{ such that}$$

$$g^n_{ij}(z_{n-1}, y_n) = z_n \, \} \, , \tag{7.4}$$

and

$$Z^{n-1}_{ij}(y_n, z_n) \equiv \{ \, z_{n-1} \; ; \; g^n_{ij}(z_{n-1}, y_n) = z_n \, \} \, . \tag{7.5}$$

Therefore, for $n = 2, \ldots, |I_{ij}|$,

$$S^n_{ij}(z_n) = \{ \, (u_{i-1j_1}, \ldots, u_{i-1j_n}) \; ;$$

$$g^n_{ij}(G^{n-1}_{ij}(u_{i-1j_1}, \ldots, u_{i-1j_{n-1}}), u_{i-1j_n}) = z_n \, \}$$

$$= \bigcup_{y_n \in V^n_{ij}(z_n)} \Big[\bigcup_{z_{n-1} \in Z^{n-1}_{ij}(y_n, z_n)} \{ \, (u_{i-1j_1}, \ldots, u_{i-1j_{n-1}}) \; ;$$

$$G^{n-1}_{ij}(u_{i-1j_1}, \ldots, u_{i-1j_{n-1}}) = z_{n-1} \, \} \times \{ \, u_{i-1j_n} \; ;$$

$$u_{i-1j_n} = y_n \, \} \Big]$$

$$= \bigcup_{y_n \in V^n_{ij}(z_n)} \Big[\bigcup_{z_{n-1} \in Z^{n-1}_{ij}(y_n, z_n)} S^{n-1}_{ij}(z_{n-1}) \times S_{i-1j_n}(y_n) \Big] \, .$$

Consequently, since the separation of F_{ij} is monotone, for $i = 2, \ldots, |I_{ij}|$,

$$P^n_{ij}(z_n) = \max \{ \, F^n_{ij}(F_{i-1j_1}, \ldots, F_{i-1j_n}) \; |$$

$$(G_{i-1j_1}, \ldots, G_{i-1j_n}) \in S^n_{ij}(z_n) \, \}$$

$$= \max \{ f_{ij}^{n}(F_{ij}^{n-1}, F_{i-1j_n}) \mid (G_{i-1j_1}, \ldots, G_{i-1j_n})$$

$$\in \bigcup_{y_n \in V_{ij}^{n}(z_n)} [\bigcup_{z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n)} S_{ij}^{n-1}(z_{n-1}) \times S_{i-1j_n}(y_n)] \}$$

$$= \max \{ f_{ij}^{n}(\max\{ F_{ij}^{n-1} \mid (G_{i-1j_1}, \ldots, G_{i-1j_n})$$

$$\in \bigcup_{z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n)} S_{ij}^{n-1}(z_{n-1}) \},$$

$$\max\{ F_{i-1j_n} \mid G_{i-1j_n} \in S_{i-1j_n}(y_n) \}) \mid y_n \in V_{ij}^{n}(z_n) \} ;$$

moreover,

$$\max\{ F_{ij}^{n-1} \mid (G_{i-1j_1}, \ldots, G_{i-1j_{n-1}}) \in \bigcup_{z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n)} S_{ij}^{n-1}(z_{n-1}) \}$$

$$= \max [\max \{ F_{ij}^{n-1} \mid (G_{i-1j_1}, \ldots, G_{i-1j_{n-1}}) \in S_{ij}^{n-1}(z_{n-1}) \} \mid$$

$$z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n)]$$

$$= \max [P_{ij}^{n-1}(z_{n-1}) \mid z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n)] ;$$

hence,

$$P_{ij}^{n}(z_n) = \max \{ f_{ij}^{n}(\max\{ P_{ij}^{n-1}(z_{n-1}) \mid z_{n-1} \in Z_{ij}^{n-1}(y_n, z_n) \},$$

$$P_{i-1j_n}(y_n)) \mid y_n \in V_{ij}^{n}(z_n) \} , \tag{7.6}$$

where for $j = 1, \ldots, N$,

$$P_{1j}(z) \equiv \max \{ F_{ij}(x_j) \mid G_{1j}(x_j) = z \quad \text{and} \quad x_j \in X_j \} \tag{7.7}$$

and P^n_{ij}, P_{i-1j_n}, V^n_{ij} and Z^{n-1}_{ij} are given by (7.2) through (7.5). This completes the proof. □

Equation (7.6) is the recursive functional equation of dynamic programming. Denote by $\bar{y}^n_{ij}(z_n)$ the value of y_n that maximizes f^n_{ij} in (7.6) and by $\bar{z}^{n-1}_{ij}(z_n)$ the value of z_{n-1} that maximizes P^{n-1}_{ij} for $\bar{y}^n_{ij}(z_n)$ in (7.6). Moreover, denote by $x^*_j(z)$ the value of x_j that maximizes F_{1j} in (7.7). This recursive relation then yields the following procedure to solve the principal problem (7.1).

Algorithm.

Step 1: Solve the jth subproblem for $j=1,\ldots,N$ in the first level $P_{1j}(z)$ defined by (7.7) for every z belonging to the region of G_{1j}.

Step 2: Set $i = 2$.

Step 3: Put for $j=1,\ldots,n_i$,

$$P^1_{ij}(z_1) = P_{i-1j_1}(z_1) ,$$

where P_{i-1j_1} was already obtained.

Step 4: Set $n = 2$.

Step 5: Solve $P^n_{ij}(z_n)$ for every z_n belonging to the region of G^n_{ij} by the recursive relation (7.6), where $P^{n-1}_{ij}(z_{n-1})$ and $P_{i-1j_n}(y_n)$ were already obtained.

Step 6: Set $n = n+1$ and if $n > |I_{ij}|$, then go to Step 7: otherwise, go to Step 5.

Step 7: Put for $j = 1, \ldots, n_i$,

$$P_{ij}(z) = P^{|I_{ij}|}(z)$$

and set $i = i + 1$. If $i > s$, then go to Step 8; otherwise, go to Step 3.

Step 8: The optimal value P of the principal problem is obtained by $P = P_{s1}(0)$.

Step 9: Compute $\bar{z}_{s1}^{|I_{s1}|-1}(0)$ by using $\bar{y}_{s1}^{|I_{s1}|}(0)$ and compute successively $\bar{y}_{s1}^{n}(\bar{z}_{s1}^{n})$, $n = 1, \ldots, |I_{ij}| - 1$.

Step 10: Put $\bar{z}_{ij_n} = \bar{y}_{i+1j}^{n}$ and compute successively $\bar{y}_{ij}^{n}(\bar{z}_{ij}^{n})$, $i = 2, \ldots, s-1$, $j = 1, \ldots, n_i$, $n = 1, \ldots, |I_{ij}|$.

Step 11: Put $\bar{z}_{1j_n} = \bar{y}_{2j}^{n}$ and determine the optimal solution of the principal problem as $(x_1^*(\bar{z}_{11}), x_2^*(\bar{z}_{12}), \ldots, x_N^*(\bar{z}_{1N}))$.

Notice that, in general, if the principal problem is not disjointly decomposable, the above computational procedure does not always give the optimal solution. Because if for some i , j , and k both F_{ij} , G_{ij} and F_{ik} and G_{ik} depend on the same variable, say, x_n , then it is not necessarily guaranteed that the optimal value of x_n derived from P_{ij} coinceides with that from P_{ik} . However, this inconsistency can be overcome by introducing additional variables as stated in Section 7.2. Then the principal problem may be reformulated as a problem having $\sum_{j=1}^{N} \sigma_{1j}$ variables and $M +$

$\sum_{j=1}^{N}(\sigma_{1j}-1)$ dimensional constraint function, since, for each j, we must add $(\sigma_{1j}-1)$ independent equalities among

$$x_j^{(1)} = x_j^{(2)} = \dots = x_1^{(\sigma_{1j})}$$

to the original constraints. It is easily verified that the left hand side of each independent equality given by

$$x_j^{(k)} - x_j^{(k')} = 0$$

is disjointly decomposable into any structure. Let the augmented constraints including these equalities be $\bar{G} = 0$; then $\bar{G}$ is disjointly decomposable. Clearly, the objective function $\bar{F}$, in which the original variables are replaced by $x_j^{(k)}$, is also disjointly decomposable. Thus we obtain the following corollary of Theorem 7.1.

<u>Corollary</u>. If the decomposition of the principal problem is not disjoint, then the original problem can be reduced to the following equivalent problem, which is disjointly decomposable, by adding some appropriate variables and constraints:

$$\begin{array}{ll} \text{maximize} & \bar{F}(x_1^{(1)},\dots,x_1^{(\sigma_{11})},\dots,x_N^{(1)},\dots,x_N^{(\sigma_{1N})}) \\ \text{subject to} & \bar{G}(x_1^{(1)},\dots,x_1^{(\sigma_{11})},\dots,x_N^{(1)},\dots,x_N^{(\sigma_{1N})}) = 0 , \\ \text{and} & x_n^{(k)} \in X_n , \quad n=1,\dots,N , \ k=1,\dots,\sigma_{1n} . \end{array}$$

If for this problem, assumptions in Theorem 7.1 are satisfied, then the problem can be decomposed into subproblems by dynamic programming.

7.4 Application to a Quadratic Fractional Programming Problem

In this section, a simple example is given to illustrate the computational procedure presented in the previous section. The principal problem to be solved is the quadratic fractional programming problem of the following type, which is two-level-decomposable and has three subproblems in the second level:

minimize

$$\frac{(\langle x_1, A_1 x_1\rangle + \langle b_1, x_1\rangle + c_1)(\langle x_2, A_2 x_2\rangle + \langle b_2, x_2\rangle + c_2)}{(\langle x_3, A_3 x_3\rangle + \langle b_3, x_3\rangle + c_3)}$$

subject to

$$\begin{aligned} \langle d_1, x_1\rangle + \langle d_2, x_2\rangle + \langle d_3, x_3\rangle &= 1 , \\ \langle e_1, x_1\rangle &= 1 , \\ \langle e_2, x_2\rangle &= 1 , \\ \langle e_3, x_3\rangle &= 1 , \end{aligned}$$

and

$$x_1 \geqq 0 , \quad x_2 \geqq 0 , \quad x_3 \geqq 0 ,$$

where for $n = 1, 2, 3$, x_n, b_n, d_n and e_n are k_n-dimensional vectors, c_n is a real number, A_n is a $k_n \times k_n$-matrix. Moreover, assume that A_1 and A_2 are positive semidefinite and A_3 is negative semidefinite.

Remark 7.1. It may be noted that this problem is a minimization problem, while we have considered maximization problems in the previous section. However, it is easy to verify that the computational procedure presented in Section 7.3 is equally applicable to minimization problems by replacing max by min everywhere it appears.

Put

$$f_1(x_1) = \langle x_1, A_1 x_1 \rangle + \langle b_1, x_1 \rangle + c_1 ,$$

$$f_2(x_2) = \langle x_2, A_2 x_2 \rangle + \langle b_2, x_2 \rangle + c_2 ,$$

$$f_3(x_3) = 1 / (\langle x_3, A_3 x_3 \rangle + \langle b_3, x_3 \rangle + c_3) ,$$

$$g_1(x_1) = \langle d_1, x_1 \rangle ,$$

$$g_2(x_2) = \langle d_2, x_2 \rangle ,$$

$$g_3(x_3) = \langle d_3, x_3 \rangle ,$$

and

$$X_1 = \{ x_1 \ ; \ \langle e_1, x_1 \rangle = 1 \text{ and } x_1 \geqq 0 \} ,$$

$$X_2 = \{ x_2 \ ; \ \langle e_2, x_2 \rangle = 1 \text{ and } x_2 \geqq 0 \} ,$$

$$X_3 = \{ x_3 \ ; \ \langle e_3, x_3 \rangle = 1 \text{ and } x_3 \geqq 0 \} .$$

Then, the principal problem can be rewritten as follows:

$$\begin{array}{lll} \text{minimize} & F(x_1,x_2,x_3) \equiv f_1(x_1)\ f_2(x_2)\ f_3(x_3) & \\ \text{subject to} & G(x_1,x_2,x_3) \equiv g_1(x_1) + g_2(x_2) + g_3(x_3) = 1 & \\ \text{and} & x_n \in X_n , & n=1,2,3 . \end{array}$$

Let us put

$$F_1(x_1) \equiv f_1(x_1) ,$$

$$F_2(x_1,x_2) \equiv f_1(x_1)\ f_2(x_2)$$
$$= F_1(x_1)\ f_2(x_2)\ ,$$
$$F_3(x_1,x_2,x_3) \equiv f_1(x_1)\ f_2(x_2)\ f_3(x_3)$$
$$= F_2(x_1,x_2)\ f_3(x_3)$$
$$= F(x_1,x_2,x_3)$$

and

$$G_1(x_1) \equiv g_1(x_1)\ ,$$
$$G_2(x_1,x_2) \equiv g_1(x_1) + g_2(x_2)$$
$$= G_1(x_1) + g_2(x_2)\ ,$$
$$G_3(x_1,x_2,x_3) \equiv g_1(x_1) + g_2(x_2) + g_3(x_3)$$
$$= G_2(x_1,x_2) + g_3(x_3)$$
$$= G(x_1,x_2,x_3)\ .$$

Thus, the objective function F and the constraint function G are sequentially separable. Moreover, if each $f_n(x_n)$ is always positive for every feasible x_n, then the sequential separation of F is always monotone, and hence, dynamic programming is applicable to this problem. Subproblems are as follows:

$$p_1(y_1) = \min\ \{\ f_1(x_1)\ |\ g_1(x_1) = y_1 \ \text{ and } \ x_1 \in X_1\ \}$$
$$= \min\ \{\ \langle x_1, A_1 x_1\rangle + \langle b_1, x_1\rangle + c_1\ |$$
$$\langle d_1, x_1\rangle = y_1\ ,\ \langle e_1, x_1\rangle = 1 \ \text{ and } \ x_1 \geqq 0\ \}\ ,$$

$$p_2(y_2) = \min \{ f_2(x_2) \mid g_2(x_2) = y_2 \text{ and } x_2 \in X_2 \}$$

$$= \min \{ \langle x_2, A_2 x_2\rangle + \langle b_2, x_2\rangle + c_2 \mid$$

$$\langle d_2, x_2\rangle = y_2 \ , \ \langle e_2, x_2\rangle = 1 \text{ and } x_2 \geqq 0 \} \ ,$$

$$p_3(y_3) = \min \{ f_3(x_3) \mid g_3(x_3) = y_3 \text{ and } x_3 \in X_3 \}$$

$$= \min \{ 1 / (\langle x_3, A_3 x_3\rangle + \langle b_3, x_3\rangle + c_3) \mid$$

$$\langle d_3, x_3\rangle = y_3 \ , \ \langle e_3, x_3\rangle = 1 \text{ and } x_3 \geqq 0 \}$$

$$= 1 / \max \{ \langle x_3, A_3 x_3\rangle + \langle b_3, x_3\rangle + c_3 \mid$$

$$\langle d_3, x_3\rangle = y_3 \ , \ \langle e_3, x_3\rangle = 1 \text{ and } x_3 \geqq 0 \} \ .$$

These problems can be solved by parametric quadratic programming methods [W4]. Put for $n = 1, 2, 3$ and $0 \leqq z_n \leqq 1$,

$$P_n(z_n) \equiv \min \{ F_n(x_1,\ldots,x_n) \mid G_n(x_1,\ldots,x_n) = z_n \text{ and } x_k \in X_k \ , \ k=1,\ldots,n \} \ .$$

Then for $n = 2, 3$,

$$P_n(z_n) = \min \{ P_{n-1}(z_n - y_n)\, p_n(y_n) \mid 0 \leqq y_n \leqq z_n \} \ ,$$

and the optimal value of the principal problem is given by $P_3(1)$.

In the case that $c_1 = c_2 = 0$, $c_3 = 5$, $b_1 = (2,0,3,0)$, $b_2 = (1,1,0,1)$, $b_3 = (0,0,-1)$, $d_1 = (1/2,0,1,3/2)$, $d_2 = (3,0,3,0)$, $d_3 = (2,0,1)$, $e_1 = (1,3/2,0,1/2)$, $e_2 = (0,2,0,5)$, $e_3 = (0,1,2)$, and

$$A_1 = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \qquad A_2 = \begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{3}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 1 \end{pmatrix} ,$$

$$A_3 = \begin{pmatrix} -\frac{1}{2} & -1 & 0 \\ -1 & -2 & 0 \\ 0 & 0 & -\frac{1}{2} \end{pmatrix} ,$$

the optimal value of the principal problem is 0.010923 , and the optimal solution is $x_1^* = (0 , 0.59556 , 0 , 0.21333)$, $x_2^* = (0 , 0 , 0.04333 , 0.2)$, $x_3^* = (0.05506 , 0.12022 , 0.43989)$. The sequences $P_n(z)$, $\bar{y}_n(z)$, and $\bar{x}_n(z)$, $n = 1 , 2 , 3$, are given in Tables 7.1 - 7.3, where $\bar{y}_n(z)$ is the value of y_n that maximizes $P_n(z)$ and $\bar{x}_n(z)$ is the value of x_n that gives $p_n(\bar{y}_n(z))$ . . The computation time is 5 seconds on a FACOM 230-75.

Table 7.1

$P_1(z)$, $\bar{y}_1(z)$, and $\bar{x}_1(z)$

z	$P_1(z)$	$\bar{y}_1(z)$	$\bar{x}_{11}(z)$	$\bar{x}_{12}(z)$	$\bar{x}_{13}(z)$	$\bar{x}_{14}(z)$
0.01	0.22077	0.01000	0.0	0.66444	0.0	0.00667
0.02	0.21936	0.02000	0.0	0.66222	0.0	0.01333
0.03	0.21800	0.03000	0.0	0.66000	0.0	0.02000
0.04	0.21669	0.04000	0.0	0.65778	0.0	0.02667
0.05	0.21543	0.05000	0.0	0.65556	0.0	0.03333
0.06	0.21422	0.06000	0.0	0.65333	0.0	0.04000
0.07	0.21306	0.07000	0.0	0.65111	0.0	0.04667
0.08	0.21195	0.08000	0.0	0.64889	0.0	0.05333
0.09	0.21089	0.09000	0.0	0.64667	0.0	0.06000
0.10	0.20988	0.10000	0.0	0.64444	0.0	0.06667
0.11	0.20891	0.11000	0.0	0.64222	0.0	0.07333
0.12	0.20800	0.12000	0.0	0.64000	0.0	0.08000
0.13	0.20714	0.13000	0.0	0.63778	0.0	0.08667
0.14	0.20632	0.14000	0.0	0.63556	0.0	0.09333
0.15	0.20556	0.15000	0.0	0.63333	0.0	0.10000
0.16	0.20484	0.16000	0.0	0.63111	0.0	0.10667
0.17	0.20417	0.17000	0.0	0.62889	0.0	0.11333
0.18	0.20356	0.18000	0.0	0.62667	0.0	0.12000
0.19	0.20299	0.19000	0.0	0.62444	0.0	0.12667
0.20	0.20247	0.20000	0.0	0.62222	0.0	0.13333
0.21	0.20200	0.21000	0.0	0.62000	0.0	0.14000
0.22	0.20158	0.22000	0.0	0.61778	0.0	0.14667
0.23	0.20121	0.23000	0.0	0.61556	0.0	0.15333
0.24	0.20089	0.24000	0.0	0.61333	0.0	0.16000
0.25	0.20062	0.25000	0.0	0.61111	0.0	0.16667
0.26	0.20040	0.26000	0.0	0.60889	0.0	0.17333
0.27	0.20022	0.27000	0.0	0.60667	0.0	0.18000
0.28	0.20010	0.28000	0.0	0.60444	0.0	0.18667
0.29	0.20002	0.29000	0.0	0.60222	0.0	0.19333
0.30	0.20000	0.30000	0.0	0.60000	0.0	0.20000
0.31	0.20002	0.31000	0.0	0.59778	0.0	0.20667
0.32	0.20010	0.32000	0.0	0.59556	0.0	0.21333
0.33	0.20022	0.33000	0.0	0.59333	0.0	0.22000
0.34	0.20040	0.34000	0.0	0.59111	0.0	0.22667
0.35	0.20062	0.35000	0.0	0.58889	0.0	0.23333
0.36	0.20089	0.36000	0.0	0.58667	0.0	0.24000
0.37	0.20121	0.37000	0.0	0.58444	0.0	0.24667
0.38	0.20158	0.38000	0.0	0.58222	0.0	0.25333
0.39	0.20200	0.39000	0.0	0.58000	0.0	0.26000
0.40	0.20247	0.40000	0.0	0.57778	0.0	0.26667
0.41	0.20299	0.41000	0.0	0.57556	0.0	0.27333
0.42	0.20356	0.42000	0.0	0.57333	0.0	0.28000
0.43	0.20417	0.43000	0.0	0.57111	0.0	0.28667
0.44	0.20484	0.44000	0.0	0.56889	0.0	0.29333
0.45	0.20556	0.45000	0.0	0.56667	0.0	0.30000
0.46	0.20632	0.46000	0.0	0.56444	0.0	0.30667
0.47	0.20714	0.47000	0.0	0.56222	0.0	0.31333
0.48	0.20800	0.48000	0.0	0.56000	0.0	0.32000
0.49	0.20891	0.49000	0.0	0.55778	0.0	0.32667

Table continued

Table 7.1 (*continued*)

z	$P_1(z)$	$\bar{y}_1(z)$	$\bar{x}_{11}(z)$	$\bar{x}_{12}(z)$	$\bar{x}_{13}(z)$	$\bar{x}_{14}(z)$
0.50	0.20988	0.50000	0.0	0.55556	0.0	0.33333
0.51	0.21089	0.51000	0.0	0.55333	0.0	0.34000
0.52	0.21195	0.52000	0.0	0.55111	0.0	0.34667
0.53	0.21306	0.53000	0.0	0.54889	0.0	0.35333
0.54	0.21422	0.54000	0.0	0.54667	0.0	0.36000
0.55	0.21543	0.55000	0.0	0.54444	0.0	0.36667
0.56	0.21669	0.56000	0.0	0.54222	0.0	0.37333
0.57	0.21800	0.57000	0.0	0.54000	0.0	0.38000
0.58	0.21936	0.58000	0.0	0.53778	0.0	0.38667
0.59	0.22077	0.59000	0.0	0.53556	0.0	0.39333
0.60	0.22222	0.60000	0.0	0.53333	0.0	0.40000
0.61	0.22373	0.61000	0.0	0.53111	0.0	0.40667
0.62	0.22528	0.62000	0.0	0.52889	0.0	0.41333
0.63	0.22689	0.63000	0.0	0.52667	0.0	0.42000
0.64	0.22854	0.64000	0.0	0.52444	0.0	0.42667
0.65	0.23025	0.65000	0.0	0.52222	0.0	0.43333
0.66	0.23200	0.66000	0.0	0.52000	0.0	0.44000
0.67	0.23380	0.67000	0.0	0.51778	0.0	0.44667
0.68	0.23565	0.68000	0.0	0.51556	0.0	0.45333
0.69	0.23756	0.69000	0.0	0.51333	0.0	0.46000
0.70	0.23951	0.70000	0.0	0.51111	0.0	0.46667
0.71	0.24151	0.71000	0.0	0.50889	0.0	0.47333
0.72	0.24356	0.72000	0.0	0.50667	0.0	0.48000
0.73	0.24565	0.73000	0.0	0.50444	0.0	0.48667
0.74	0.24780	0.74000	0.0	0.50222	0.0	0.49333
0.75	0.25000	0.75000	0.0	0.50000	0.0	0.50000
0.76	0.25225	0.76000	0.0	0.49778	0.0	0.50667
0.77	0.25454	0.77000	0.0	0.49556	0.0	0.51333
0.78	0.25689	0.78000	0.0	0.49333	0.0	0.52000
0.79	0.25928	0.79000	0.0	0.49111	0.0	0.52667
0.80	0.26173	0.80000	0.0	0.48889	0.0	0.53333
0.81	0.26422	0.81000	0.0	0.48667	0.0	0.54000
0.82	0.26677	0.82000	0.0	0.48444	0.0	0.54667
0.83	0.26936	0.83000	0.0	0.48222	0.0	0.55333
0.84	0.27200	0.84000	0.0	0.48000	0.0	0.56000
0.85	0.27469	0.85000	0.0	0.47778	0.0	0.56667
0.86	0.27743	0.86000	0.0	0.47556	0.0	0.57333
0.87	0.28022	0.87000	0.0	0.47333	0.0	0.58000
0.88	0.28306	0.88000	0.0	0.47111	0.0	0.58667
0.89	0.28595	0.89000	0.0	0.46889	0.0	0.59333
0.90	0.28889	0.90000	0.0	0.46667	0.0	0.60000
0.91	0.29188	0.91000	0.0	0.46444	0.0	0.60667
0.92	0.29491	0.92000	0.0	0.46222	0.0	0.61333
0.93	0.29800	0.93000	0.0	0.46000	0.0	0.62000
0.94	0.30114	0.94000	0.0	0.46778	0.0	0.62667
0.95	0.30432	0.95000	0.0	0.45556	0.0	0.63333
0.96	0.30756	0.96000	0.0	0.45333	0.0	0.64000
0.97	0.31084	0.97000	0.0	0.45111	0.0	0.64667
0.98	0.31417	0.98000	0.0	0.44889	0.0	0.65333
0.99	0.31756	0.99000	0.0	0.44667	0.0	0.66000
1.00	0.32099	1.00000	0.0	0.44444	0.0	0.66667

Table 7.2

$P_2(z)$, $\bar{y}_2(z)$, and $\bar{x}_2(z)$

z	$P_2(z)$	$\bar{y}_2(z)$	$\bar{x}_{21}(z)$	$\bar{x}_{22}(z)$	$\bar{x}_{23}(z)$	$\bar{x}_{24}(z)$
0.01	0.05298	0.01000	0.0	0.0	0.00333	0.20000
0.02	0.05298	0.01000	0.0	0.0	0.00333	0.20000
0.03	0.05265	0.01000	0.0	0.0	0.00333	0.20000
0.04	0.05232	0.01000	0.0	0.0	0.00333	0.20000
0.05	0.05201	0.01000	0.0	0.0	0.00333	0.20000
0.06	0.05170	0.01000	0.0	0.0	0.00333	0.20000
0.07	0.05141	0.01000	0.0	0.0	0.00333	0.20000
0.08	0.05114	0.01000	0.0	0.0	0.00333	0.20000
0.09	0.05087	0.01000	0.0	0.0	0.00333	0.20000
0.10	0.05061	0.01000	0.0	0.0	0.00333	0.20000
0.11	0.05037	0.01000	0.0	0.0	0.00333	0.20000
0.12	0.05014	0.01000	0.0	0.0	0.00333	0.20000
0.13	0.04992	0.01000	0.0	0.0	0.00333	0.20000
0.14	0.04971	0.01000	0.0	0.0	0.00333	0.20000
0.15	0.04952	0.01000	0.0	0.0	0.00333	0.20000
0.16	0.04933	0.01000	0.0	0.0	0.00333	0.20000
0.17	0.04916	0.01000	0.0	0.0	0.00333	0.20000
0.18	0.04900	0.01000	0.0	0.0	0.00333	0.20000
0.19	0.04885	0.01000	0.0	0.0	0.00333	0.20000
0.20	0.04872	0.01000	0.0	0.0	0.00333	0.20000
0.21	0.04859	0.01000	0.0	0.0	0.00333	0.20000
0.22	0.04848	0.01000	0.0	0.0	0.00333	0.20000
0.23	0.04838	0.01000	0.0	0.0	0.00333	0.20000
0.24	0.04829	0.01000	0.0	0.0	0.00333	0.20000
0.25	0.04821	0.01000	0.0	0.0	0.00333	0.20000
0.26	0.04815	0.01000	0.0	0.0	0.00333	0.20000
0.27	0.04810	0.01000	0.0	0.0	0.00333	0.20000
0.28	0.04805	0.01000	0.0	0.0	0.00333	0.20000
0.29	0.04802	0.01000	0.0	0.0	0.00333	0.20000
0.30	0.04801	0.01000	0.0	0.0	0.00333	0.20000
0.31	0.04800	0.01000	0.0	0.0	0.00333	0.20000
0.32	0.04800	0.03000	0.0	0.0	0.00667	0.20000
0.33	0.04801	0.03000	0.0	0.0	0.01000	0.20000
0.34	0.04802	0.03000	0.0	0.0	0.01000	0.20000
0.35	0.04802	0.04000	0.0	0.0	0.01333	0.20000
0.36	0.04803	0.05000	0.0	0.0	0.01667	0.20000
0.37	0.04805	0.06000	0.0	0.0	0.02000	0.20000
0.38	0.04806	0.07000	0.0	0.0	0.02333	0.20000
0.39	0.04808	0.08000	0.0	0.0	0.02667	0.20000
0.40	0.04809	0.08000	0.0	0.0	0.02667	0.20000
0.41	0.04811	0.09000	0.0	0.0	0.03000	0.20000
0.42	0.04813	0.10000	0.0	0.0	0.03333	0.20000
0.43	0.04816	0.11000	0.0	0.0	0.03667	0.20000
0.44	0.04818	0.12000	0.0	0.0	0.04000	0.20000
0.45	0.04821	0.13000	0.0	0.0	0.04333	0.20000
0.46	0.04824	0.13000	0.0	0.0	0.04333	0.20000
0.47	0.04827	0.14000	0.0	0.0	0.04667	0.20000
0.48	0.04830	0.15000	0.0	0.0	0.05000	0.20000
0.49	0.04834	0.16000	0.0	0.0	0.05333	0.20000

Table continued

Table 7.2 (*continued*)

z	$P_2(z)$	$\bar{y}_2(z)$	$\bar{x}_{21}(z)$	$\bar{x}_{22}(z)$	$\bar{x}_{23}(z)$	$\bar{x}_{24}(z)$
0.50	0.04837	0.17000	0.0	0.0	0.05667	0.20000
0.51	0.04841	0.18000	0.0	0.0	0.06000	0.20000
0.52	0.04845	0.19000	0.0	0.0	0.06333	0.20000
0.53	0.04850	0.19000	0.0	0.0	0.06333	0.20000
0.54	0.04854	0.20000	0.0	0.0	0.06667	0.20000
0.55	0.04859	0.21000	0.0	0.0	0.07000	0.20000
0.56	0.04863	0.22000	0.0	0.0	0.07333	0.20000
0.57	0.04868	0.23000	0.0	0.0	0.07667	0.20000
0.58	0.04874	0.24000	0.0	0.0	0.08000	0.20000
0.59	0.04879	0.24000	0.0	0.0	0.08000	0.20000
0.60	0.04884	0.25000	0.0	0.0	0.08333	0.20000
0.61	0.04890	0.26000	0.0	0.0	0.08667	0.20000
0.62	0.04896	0.27000	0.0	0.0	0.09000	0.20000
0.63	0.04902	0.28000	0.0	0.0	0.09333	0.20000
0.64	0.04909	0.29000	0.0	0.0	0.09667	0.20000
0.65	0.04915	0.30000	0.0	0.0	0.10000	0.20000
0.66	0.04922	0.30000	0.0	0.0	0.10000	0.20000
0.67	0.04929	0.31000	0.0	0.0	0.10333	0.20000
0.68	0.04936	0.32000	0.0	0.0	0.10667	0.20000
0.69	0.04943	0.33000	0.0	0.0	0.11000	0.20000
0.70	0.04950	0.34000	0.0	0.0	0.11333	0.20000
0.71	0.04958	0.35000	0.0	0.0	0.11667	0.20000
0.72	0.04966	0.35000	0.0	0.0	0.11667	0.20000
0.73	0.04974	0.36000	0.0	0.0	0.12000	0.20000
0.74	0.04982	0.37000	0.0	0.0	0.12333	0.20000
0.75	0.04990	0.38000	0.0	0.0	0.12667	0.20000
0.76	0.04999	0.39000	0.0	0.0	0.13000	0.20000
0.77	0.05008	0.40000	0.0	0.0	0.13333	0.20000
0.78	0.05017	0.41000	0.0	0.0	0.13667	0.20000
0.79	0.05026	0.41000	0.0	0.0	0.13667	0.20000
0.80	0.05035	0.42000	0.0	0.0	0.14000	0.20000
0.81	0.05045	0.43000	0.0	0.0	0.14333	0.20000
0.82	0.05055	0.44000	0.0	0.0	0.14667	0.20000
0.83	0.05065	0.45000	0.0	0.0	0.15000	0.20000
0.84	0.05075	0.46000	0.0	0.0	0.15333	0.20000
0.85	0.05085	0.47000	0.0	0.0	0.15667	0.20000
0.86	0.05096	0.47000	0.0	0.0	0.15667	0.20000
0.87	0.05107	0.48000	0.0	0.0	0.16000	0.20000
0.88	0.05117	0.49000	0.0	0.0	0.16333	0.20000
0.89	0.05129	0.50000	0.0	0.0	0.16667	0.20000
0.90	0.05140	0.51000	0.0	0.0	0.17000	0.20000
0.91	0.05151	0.52000	0.0	0.0	0.17333	0.20000
0.92	0.05163	0.53000	0.0	0.0	0.17667	0.20000
0.93	0.05175	0.53000	0.0	0.0	0.17667	0.20000
0.94	0.05187	0.54000	0.0	0.0	0.18000	0.20000
0.95	0.05200	0.55000	0.0	0.0	0.18333	0.20000
0.96	0.05212	0.56000	0.0	0.0	0.18667	0.20000
0.97	0.05225	0.57000	0.0	0.0	0.19000	0.20000
0.98	0.05238	0.58000	0.0	0.0	0.19333	0.20000
0.99	0.05251	0.59000	0.0	0.0	0.19667	0.20000
1.00	0.05264	0.60000	0.0	0.0	0.20000	0.20000

Table 7.3

$P_3(z)$, $\bar{y}_3(z)$, and $\bar{x}_3(z)$

z	$P_3(z)$	$\bar{y}_3(z)$	$\bar{x}_{31}(z)$	$\bar{x}_{32}(z)$	$\bar{x}_{33}(z)$
0.01	0.01726	0.01000	0.0	0.98000	0.01000
0.02	0.01726	0.01000	0.0	0.98000	0.01000
0.03	0.01689	0.02000	0.0	0.96000	0.02000
0.04	0.01689	0.02000	0.0	0.96000	0.02000
0.05	0.01655	0.03000	0.0	0.94000	0.03000
0.06	0.01622	0.04000	0.0	0.92000	0.04000
0.07	0.01563	0.06000	0.0	0.88000	0.06000
0.08	0.01563	0.06000	0.0	0.88000	0.06000
0.09	0.01537	0.07000	0.0	0.86000	0.07000
0.10	0.01511	0.08000	0.0	0.84000	0.08000
0.11	0.01488	0.09000	0.0	0.82000	0.09000
0.12	0.01466	0.10000	0.0	0.80000	0.10000
0.13	0.01445	0.11000	0.0	0.78000	0.11000
0.14	0.01407	0.13000	0.0	0.74000	0.13000
0.15	0.01389	0.14000	0.0	0.72000	0.14000
0.16	0.01373	0.15000	0.0	0.70000	0.15000
0.17	0.01358	0.16000	0.0	0.68000	0.16000
0.18	0.01343	0.17000	0.0	0.66000	0.17000
0.19	0.01330	0.18000	0.0	0.64000	0.18000
0.20	0.01317	0.19000	0.0	0.62000	0.19000
0.21	0.01305	0.20000	0.0	0.60000	0.20000
0.22	0.01294	0.21000	0.0	0.58000	0.21000
0.23	0.01283	0.22000	0.0	0.56000	0.22000
0.24	0.01274	0.23000	0.0	0.54000	0.23000
0.25	0.01274	0.23000	0.0	0.54000	0.23000
0.26	0.01264	0.24000	0.0	0.52000	0.24000
0.27	0.01256	0.25000	0.0	0.50000	0.25000
0.28	0.01248	0.25000	0.0	0.50000	0.25000
0.29	0.01240	0.26000	0.0	0.48000	0.26000
0.30	0.01232	0.26000	0.0	0.48000	0.26000
0.31	0.01225	0.26000	0.0	0.48000	0.26000
0.32	0.01218	0.27000	0.0	0.46000	0.27000
0.33	0.01211	0.27000	0.0	0.46000	0.27000
0.34	0.01204	0.27000	0.0	0.46000	0.27000
0.35	0.01197	0.27000	0.0	0.46000	0.27000
0.36	0.01191	0.28000	0.0	0.44000	0.28000
0.37	0.01185	0.28000	0.0	0.44000	0.28000
0.38	0.01179	0.28000	0.0	0.44000	0.28000
0.39	0.01173	0.29000	0.0	0.42000	0.29000
0.40	0.01167	0.29000	0.0	0.42000	0.29000
0.41	0.01162	0.29000	0.0	0.42000	0.29000
0.42	0.01157	0.30000	0.0	0.40000	0.30000
0.43	0.01152	0.30000	0.0	0.40000	0.30000
0.44	0.01147	0.31000	0.0	0.38000	0.31000
0.45	0.01142	0.31000	0.0	0.38000	0.31000
0.46	0.01138	0.31000	0.0	0.38000	0.31000
0.47	0.01133	0.32000	0.0	0.36000	0.32000
0.48	0.01129	0.32000	0.0	0.36000	0.32000
0.49	0.01125	0.32000	0.0	0.36000	0.32000

Table continued

Table 7.3 (*continued*)

z	$P_3(z)$	$\bar{y}_3(z)$	$\bar{x}_{31}(z)$	$\bar{x}_{32}(z)$	$\bar{x}_{33}(z)$
0.50	0.01121	0.33000	0.0	0.34000	0.33000
0.51	0.01118	0.33000	0.0	0.34000	0.33000
0.52	0.01114	0.33000	0.0	0.34000	0.33000
0.53	0.01111	0.34000	0.0	0.32000	0.34000
0.54	0.01108	0.34000	0.0	0.32000	0.34000
0.55	0.01105	0.35000	0.0	0.30000	0.35000
0.56	0.01102	0.35000	0.0	0.30000	0.35000
0.57	0.01100	0.35000	0.0	0.30000	0.35000
0.58	0.01097	0.36000	0.0	0.28000	0.36000
0.59	0.01095	0.36000	0.0	0.28000	0.36000
0.60	0.01093	0.36000	0.0	0.28000	0.36000
0.61	0.01091	0.37000	0.0	0.26000	0.37000
0.62	0.01089	0.37000	0.0	0.26000	0.37000
0.63	0.01088	0.38000	0.0	0.24000	0.38000
0.64	0.01086	0.38000	0.0	0.24000	0.38000
0.65	0.01085	0.38000	0.0	0.24000	0.38000
0.66	0.01084	0.39000	0.0	0.22000	0.39000
0.67	0.01083	0.39000	0.0	0.22000	0.39000
0.68	0.01082	0.40000	0.0	0.20000	0.40000
0.69	0.01082	0.40000	0.0	0.20000	0.40000
0.70	0.01081	0.40000	0.0	0.20000	0.40000
0.71	0.01081	0.41000	0.0	0.18000	0.41000
0.72	0.01081	0.41000	0.0	0.18000	0.41000
0.73	0.01081	0.41000	0.0	0.18000	0.41000
0.74	0.01081	0.41000	0.0	0.18000	0.41000
0.75	0.01081	0.41000	0.0	0.18000	0.41000
0.76	0.01081	0.42000	0.0	0.16000	0.42000
0.77	0.01081	0.42000	0.0	0.16000	0.42000
0.78	0.01082	0.42000	0.0	0.16000	0.42000
0.79	0.01082	0.42000	0.0	0.16000	0.42000
0.80	0.01082	0.42000	0.0	0.16000	0.42000
0.81	0.01083	0.42000	0.0	0.16000	0.42000
0.82	0.01083	0.43000	0.00382	0.15528	0.42236
0.83	0.01083	0.43000	0.00382	0.15528	0.42236
0.84	0.01084	0.44000	0.00809	0.15236	0.42382
0.85	0.01084	0.44000	0.00809	0.15236	0.42382
0.86	0.01085	0.45000	0.01236	0.14944	0.42528
0.87	0.01085	0.46000	0.01663	0.14652	0.42674
0.88	0.01086	0.47000	0.02090	0.14360	0.42820
0.89	0.01086	0.47000	0.02090	0.14360	0.42820
0.90	0.01087	0.48000	0.02517	0.14067	0.42966
0.91	0.01087	0.49000	0.02944	0.13775	0.43112
0.92	0.01088	0.49000	0.02944	0.13775	0.43112
0.93	0.01088	0.50000	0.03371	0.13483	0.43258
0.94	0.01089	0.51000	0.03798	0.13191	0.43404
0.95	0.01089	0.52000	0.04225	0.12899	0.43551
0.96	0.01090	0.52000	0.04225	0.12899	0.43551
0.97	0.01090	0.53000	0.04652	0.12607	0.43697
0.98	0.01091	0.54000	0.05079	0.12315	0.43843
0.99	0.01092	0.55000	0.05506	0.12022	0.43989
1.00	0.01092	0.55000	0.05506	0.12022	0.43989

7.5 Continuous Objective and Constraint Functions

In this section, the principal problem with a continuous objective function and continuous constraint functions is dealt with, and the decomposability and the separability of those functions are discussed. Throughout this section, it is assumed that the variables x_n in the principal problem are real numbers, and that D^N is the unit cube in R^N.

Sprecher [S5] proved the following theorem.

Theorem 7.2. Every real continuous function of N variables, $H(x_1,\ldots,x_N)$, with domain D^N, can be represented in the form

$$H(x_1,\ldots,x_N) = h[\sum_{n=1}^{N} h_n(x_n)] ,$$

where each h_n, $n=1,\ldots,N$, is a real monotone increasing function.

Note that the function h in Theorem 7.2 is, in general, discontinuous. It is easily verified that this theorem can be generalized as follows.

Corollary. Every real continuous functions $H(x_1,\ldots,x_N)$ defined on D^N can be represented in the form

$$H(x_1,\ldots,x_N) = h[\psi(h_1(x_1),\ldots,h_N(x_N))] ,$$

where each h_n, $n=1,\ldots,N$, is a real monotone increasing function and ψ is a monotonically and sequentially separable function.

From these results, we have the following theorem.

Theorem 7.3. Suppose that the objective function F and the constraint function G of the principal problem are continuous, and that $X_n = [0,1] \subset R$. Then the following is valid on D^N:

$$F(x_1,\ldots,x_N) = f[\psi(f_1(x_1),\ldots,f_N(x_N))] ,$$

$$G_m(x_1,\ldots,x_N) = g_m[\psi_m(g_{m1}(x_1),\ldots,g_{mN}(x_N))] , \quad m=1,\ldots,M ,$$

where f_n and g_{mn} are real monotone increasing functions and ψ and ψ_m are real monotonically and sequentially separable functions. Moreover, let

$$Y_m \equiv \{ y_m ; g_m(y_m) = 0 \} ,$$

and assume that the function f is monotone nondecreasing function of ψ. Then the principal problem is equivalent to the following problem:

$$\begin{aligned} \max \Big[\max \Big\{ & \psi (f_1(x_1),\ldots,f_N(x_N)) \;\Big|\; \\ & \psi_m (g_{m1}(x_1),\ldots,g_{mN}(x_N)) = y_m , \quad m=1,\ldots,M , \\ & \text{and} \quad (x_1,\ldots,x_N) \in D^N \Big\} \;\Big|\; (y_1,\ldots,y_M) \in Y_1 \times \ldots \times Y_M \Big] . \end{aligned}$$

In the above theorem, we have assumed that X_n is the unit interval in R. However, it is not difficult to see that the statement of the theorem remains valid even when X_n is any compact interval in R. From the fact that ψ and ψ_m are monotonically and sequentially separable, Theorems 7.1 and 7.3 lead to the following theorem.

Theorem 7.4. Suppose that all the assumptions in Theorem 7.3 are satisfied. Let

$$P(y) \equiv \max \{ \psi(f_1(x_1),\dots,f_N(x_N)) \mid \psi_m(g_{m1}(x_1),\dots,g_{mN}(x_N)) = y_m \ , \ m=1,\dots,M \ , \text{ and } (x_1,\dots,x_N) \in D^N \} \ ,$$

where $y = (y_1,\dots,y_M)$. Then $P(y)$ can be decomposed into sub-problems by dynamic programming. Furthermore, the optimal solution of the principal problem is found by solving the problem

$$\max \{ P(y) \mid y \in Y_1 \times \dots \times Y_M \} .$$

In particular, if each g_m is one-to-one, then each y_m can be determined uniquely, and hence, the optimal solution can be obtained by solving the $P(y)$.

7.6 Conclusion

In this chapter, the applicability of dynamic programming to nonlinear programming problems has been considered and sufficient conditions for the nonlinear programming problems to be decomposed by dynamic programming have been stated. However, due to the fact that the decomposition is not unique, some other questions may arise. Which is the best of all the possible decompositions? How can such a decomposition be accomplished? These problems require further research. The 'best' decomposition may be defined, for instance, as the one that makes the subproblems easiest to calculate, or as the one that consists of the least number of levels, depending on the circumstances.

On the other hand, in practical computation, parametric subproblems must be solved by a certain parametric programming technique such as [M11]. Moreover, additional variables introduced when the principal problem is transformed into the disjoint equivalent must be as few as possible, since the increase of the dimensionality of the constraints requires exponentially increased computation time and memory storages.

CHAPTER 8

DECOMPOSITION OF NONLINEAR CHANCE-CONSTRAINED PROGRAMMING PROBLEMS BY DYNAMIC PROGRAMMING

In this chapter, we consider a nonlinear chance-constrained programming problem which is one of the important and interesting problems in the field of probabilistic programming. In the previous chapter, we have studied the applicability of dynamic programming to ordinary nonlinear programming problems. Here we find a sufficient condition for the chance-constrained programming problem to be decomposed into subproblems and to be solved via recursive functional equations of dynamic programming.

8.1 Introduction

Mathematical programming problems under uncertainty are generally called probabilistic programming problems or stochastic programming problems [S1][V2]. There are actually various ways of formulating such problems and among others one of the commonest formulations is that of chance-constrained programming [C1][J1] [V2]. The basic idea of chance-constraints is to consider such decision variables as feasible that violate the constraints containing random variables within certain probability. Normally, the chance-constrained programming problem is solved by transforming it into its deterministic equivalent. However, the applicability of such techniques is somewhat limited, because attention is mostly restricted to the case where the constraint functions are all linear and the random variables are of normal distribution.

In this chapter, we deal with nonlinear chance-constrained programming problems from a viewpoint of dynamic programming under a very general setting. Dynamic programming, as is well known, can solve various types of optimization problems [B3][N1], but there appears no attempt to solve chance-constrained programming problems, except [G11] which proposes an approach quite different from this chapter. Here, we discuss the decomposability of a nonlinear chance-constrained programming problem into tractable subproblems by dynamic programming. In contrast with the deterministic case, however, it is impossible to establish recursive functional equations that yield an optimal solution for such problems due to difficulties in computation of probabilities. Therefore, we derive recursive functional equations of dynamic programming which give the relations between, both upper and lower, bounds of the optimal values of the subproblems. As a result, the nonlinear chance-constrained programming problems may be solved approximately. An advantage of this approach is that it enables us to handle random variables which are not necessarily normal.

This chapter is organized as follows: In Section 8.2, the problem is formally stated together with some assumptions on the problem. In Section 8.3, we state the main theorem which gives recursive functional equations of dynamic programming. In Section 8.4, a simple example is given to illustrate this approach.

8.2 The Problem and Assumptions

A chance-constrained programming problem to be considered here is a problem of the following type:

$$\text{maximize} \qquad F(f_1(x_1), \ldots, f_N(x_N)) \tag{8.1}$$

subject to

$$\Pr\{ G_m(g_{m1}(a_1, x_1), \ldots, g_{mN}(a_N, x_N)) \leqq 0 \} \geqq \alpha_m , \qquad m = 1, \ldots, M ,$$

and

$$x_n \in X_n , \qquad n = 1, \ldots, N ,$$

where x_n , $n=1,\ldots,N$, is a k_n-vector, a_n , $n=1,\ldots,N$, is an independent ℓ_n-dimensional random vector with the distribution function ϕ_n , the objective function F and the constraint functions G_m , $m = 1,\ldots,M$, are real valued functions on R^N , the real valued functions f_n , $n=1,\ldots,N$, and g_{mn} , $m=1,\ldots,M$, $n=1,\ldots,N$, are defined on $X_n \subset R^{k_n}$ and $X_n \times R^{\ell_n}$, respectively, and α_m , $m=1,\ldots,M$, is a given positive probability. Throughout this chapter, this problem is called *the principal problem*. In the case that for N-dimensional vectors c , x , N-dimensional random vectors a_m , $m=1,\ldots,M$, and random variables b_m , $m=1,\ldots,M$, the problem is of the form

$$\text{maximize} \qquad \langle c, x \rangle$$

$$\text{subject to} \qquad \Pr\{ \langle a_m, x \rangle \leqq b_m \} \geqq \alpha_m , \qquad m=1,\ldots,M ,$$

some methods of solving this problem are available [S1][V2]. In case of the problem with nonlinear objective and constraint functions, however, it seems that little attempt has been made to devise effective solution methods.

In the following, to simplify the notation, the symbol a_n denotes the ℓ_n-dimensional random vector and its value interchangeably. This will cause no confusion, since the distinction is always clear from the context. We assume that all integrals which appear in this chapter are well-defined. We also assume that every maximization problem $\max\{\cdot \mid \cdot\}$ is finitely attained unless its feasible region is empty. In the latter case, we define $\max\{\cdot \mid \emptyset\} = -\infty$ conventionally.

In the remainder of this section, we give some assumptions and notations that will be used in subsequent sections.

First we assume that the objective function F and the constraint functions G_m , $m=1,\ldots,M$, are *strongly decomposable* [M12] with real valued *separating functions* h_n , $n=1,\ldots,N$, and h_{mn} , $m=1,\ldots,M$, $n=1,\ldots,N$, respectively, which are defined on R^2 : That is, there exist real valued functions F_n , $n=1,\ldots,N$, and G_{mn} , $m=1,\ldots,M$, $n=1,\ldots,N$, defined on R^n such that

$$F_1(f_1(x_1)) = h_1(f_1(x_1)) ,$$

$$F_n(f_1(x_1),\ldots,f_n(x_n)) = h_n(F_{n-1}(f_1(x_1),\ldots,f_{n-1}(x_{n-1}),f_n(x_n)) ,$$

$$n = 2,\ldots,N ,$$

$$F_N(f_1(x_1),\ldots,f_N(x_N)) = F(f_1(x_1),\ldots,f_N(x_N)) ,$$

and for $m=1,\ldots,M$,

$$G_{m1}(g_{m1}(a_1,x_1)) = h_{m1}(g_{m1}(a_1,x_1)) ,$$

$$G_{mn}(g_{m1}(a_1,x_1),\ldots,g_{mn}(a_n,x_n)) =$$

$$h_{mn}(G_{mn-1}(g_{m1}(a_1,x_1),\ldots,g_{mn-1}(a_{n-1},x_{n-1})),g_{mn}(a_n,x_n)) ,$$

$$n = 2,\ldots, N , \tag{8.2}$$

$$G_{mN}(g_{m1}(a_1,x_1),\ldots,g_{mN}(a_N,x_N)) = G_m(g_{m1}(a_1,x_1),\ldots,g_{mN}(a_N,x_N)) .$$

Moreover, for $m = 1,\ldots, M$ and $n = 1,\ldots, N$, the separating functions h_n and h_{mn} are monotone, i.e., they are nondecreasing with respect to each argument.

Then, the *nth stage chance-constrained programming problem* is defined as follows:

$$\begin{aligned} &\text{maximize} \quad F_n(f_1(x_1),\ldots,f_n(x_n)) \\ &\text{subject to} \\ &\qquad \Pr\{ G_{mn}(g_{m1}(a_1,x_1),\ldots,g_{mn}(a_n,x_n)) \leqq \theta_{mn} \} \geqq \alpha_{mn} , \\ &\qquad\qquad m = 1,\ldots, M , \\ &\text{and} \qquad x_k \in X_k , \qquad k = 1,\ldots, n , \end{aligned}$$

where θ_{mn} are real numbers and α_{mn} are probabilities. This problem is abbreviated to the following:

$$\begin{aligned} P_n(\theta_n,\alpha_n) \equiv \max\{ &F_n(f_1(x_1),\ldots,f_n(x_n)) \mid \\ &\Pr\{ G_{mn}(g_{m1}(a_1,x_1),\ldots,g_{mn}(a_n,x_n)) \leqq \theta_{mn} \} \\ &\geqq \alpha_{mn} , \ m=1,\ldots,M \text{ and } x_k \in X_k , \ k=1,\ldots,n \} , \end{aligned}$$

where $\theta_n = (\theta_{1n},\ldots,\theta_{Mn})$ and $\alpha_n = (\alpha_{1n},\ldots,\alpha_{Mn})$. In particular, for $0 \in R^M$ and $\alpha = (\alpha_1, \ldots, \alpha_M)$, the principal problem (8.1) is given by $P_N(0, \alpha)$.

For $m = 1, \ldots, M$ and $n = 1, \ldots, N$, the sets V_{mn}, S_{mn} and S_n are defined as follows:

$$V_{mn}(x_1, \ldots, x_n, \theta_{mn}) \equiv \{ (a_1, \ldots, a_n) ;$$
$$G_{mn}(g_{m1}(a_1, x_1), \ldots, g_{mn}(a_n, x_n)) \leqq \theta_{mn} \} , \quad (8.3)$$

$$S_{mn}(\theta_{mn}, \alpha_{mn}) \equiv \{ (x_1, \ldots, x_n) ; \Pr\{ G_{mn}(g_{m1}(a_1, x_1), \ldots,$$
$$g_{mn}(a_n, x_n) \leqq \theta_{mn} \} \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k , \ k=1, \ldots, n \}, \quad (8.4)$$

and

$$S_n(\theta_n, \alpha_n) \equiv \{ (x_1, \ldots, x_n) ; \quad \Pr\{ G_{mn}(g_{m1}(a_1, x_1), \ldots,$$
$$g_{mn}(a_n, x_n)) \leqq \theta_{mn} \} \geqq \alpha_{mn} \}, \ m = 1, \ldots, m$$
$$\text{and} \quad x_k \in X_k , \ k = 1, \ldots, n \}$$
$$= \bigcap_{m=1}^{M} S_{mn}(\theta_{mn}, \alpha_{mn}) .$$

The set $V_{mn}(x_1, \ldots, x_n, \theta_{mn})$ is sometimes referred to as the *acceptance region* for the chance constraint $\Pr\{ G_{mn}(g_{m1}(a_1, x_1), \ldots, g_{mn}(a_n, x_n)) \leqq \theta_{mn} \} \geqq \alpha_{mn}$ under the decision rule $(x_1, \ldots, x_n)$ [C2]. Therefore, for $n = 1, \ldots, N$,

$$P_n(\theta_n, \alpha_n) = \max \{ F_n(f_1(x_1), \ldots, f_n(x_n)) \mid (x_1, \ldots, x_n) \in S_n(\theta_n, \alpha_n) \}$$
$$= \max \{ h_n(F_{n-1}, f_n) \mid (x_1, \ldots, x_n) \in S_n(\theta_n, \alpha_n) \} . \quad (8.5)$$

For $m = 1, \ldots, M$ and $n = 2, \ldots, N$, by (8.2) and the monotonicity of the separating functions h_{mn} of G_m, we have

$$V_{mn}(x_1,\dots,x_n,\theta_{mn}) = \{ (a_1,\dots,a_n) \; ; \; h_{mn}(G_{mn-1},g_{mn}) \leqq \theta_{mn} \}$$

$$= \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} \{ (a_1,\dots,a_n) \; ;$$

$$G_{mn-1}(g_{m1}(a_1,x_1),\dots,g_{mn-1}(a_{n-1},x_{n-1}))$$

$$\leqq \theta_{mn-1} \text{ and } g_{mn}(a_n,x_n) \leqq \delta_{mn} \}$$

$$= \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [\{(a_1,\dots,a_{n-1}) \; ; \; G_{mn-1} \leqq \theta_{mn-1}\}$$

$$\times \{ a_n \; ; \; g_{mn} \leqq \delta_{mn} \}]$$

$$= \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [V_{mn-1}(x_1,\dots,x_{n-1},\theta_{mn-1})$$

$$\times v_{mn}(x_n,\delta_{mn})] , \tag{8.6}$$

where

$$D_{mn}(\theta_{mn}) \equiv \{ \delta_{mn} \; ; \text{ there exists } \theta'_{mn-1} \text{ such that}$$

$$h_{mn}(\theta'_{mn-1},\delta_{mn}) \leqq \theta_{mn} \} ,$$

$$v_{mn}(x_n,\delta_{mn}) \equiv \{ a_n \; ; \; g_{mn}(a_n,x_n) \leqq \delta_{mn} \} \tag{8.7}$$

and for fixed θ_{mn} and $\delta_{mn} \in D_{mn}(\theta_{mn})$, θ_{mn-1} is determined as

$$\theta_{mn-1} = \sup \{ \theta'_{mn-1} \mid h_{mn}(\theta'_{mn-1},\delta_{mn}) \leqq \theta_{mn} \} ,$$

where the supremum is assumed to be finite. Note that θ_{mn-1} depends upon θ_{mn} and δ_{mn}.

It then follows from (8.3), (8.4) and (8.6) that for $n = 2,\dots,N$

and $m=1,\ldots,M$, $S_{mn}(\theta_{mn},\alpha_{mn})$ is represented in terms of the distribution functions ϕ_n as

$$S_{mn}(\theta_{mn},\alpha_{mn})$$

$$= \{ (x_1,\ldots,x_n) ; \int\cdots\int_{V_{mn}(x_1,\ldots,x_n,\theta_{mn})} d\phi_1(a_1)\ldots d\phi_n(a_n) \geqq \alpha_{mn}$$

$$\text{and } x_k \in X_k , \ k=1,\ldots,n \}$$

$$= \{ (x_1,\ldots,x_n) ; \int\cdots\int_{\underset{\delta_{mn}\in D_{mn}(\theta_{mn})}{\cup} [V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})]}$$

$$\times \, d\phi_1(a_1)\ldots d\phi_n(a_n) \geqq \alpha_{mn} \text{ and } x_k \in X_k ,$$

$$k=1,\ldots,n \} . \tag{8.8}$$

Let us define for $m=1,\ldots,M$ and $n=2,\ldots,N$,

$$D_n(\theta_n) \equiv \overset{M}{\underset{m=1}{\times}} D_{mn}(\theta_{mn}) ,$$

$$\bar{\theta}_{mn-1} \equiv \sup\{ \theta_{mn-1}(\theta_{mn},\delta_{mn}) \mid \delta_{mn} \in D_{mn}(\theta_{mn}) \} , \tag{8.9}$$

$$\bar{\theta}_{n-1} = (\bar{\theta}_{1n-1},\ldots,\bar{\theta}_{Mn-1}) ,$$

$$\bar{\delta}_{mn} \equiv \sup\{ \delta_{mn} \mid \delta_{mn} \in D_{mn}(\theta_{mn}) \} ,$$

$$\bar{\delta}_n = (\bar{\delta}_{1n},\ldots,\bar{\delta}_{Mn}) , \tag{8.10}$$

$$\underline{S}_{mn}(\theta_{mn},\alpha_{mn}) \equiv \bigcup_{\delta_{mn}\in D_{mn}(\theta_{mn})} \{ (x_1,\ldots,x_n) ;$$

$$\int\cdots\int_{V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1}) \quad)$$

$$\times \int_{v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k \ , \ k=1,\ldots,n \} \ , \qquad (8.11)$$

$$\underline{S}_n(\theta_n,\alpha_n) \equiv \bigcap_{m=1}^{M} \underline{S}_{mn}(\theta_{mn},\alpha_{mn}) \ , \qquad (8.12)$$

$$\underline{P}_n(\theta_n,\alpha_n) \equiv \max\{ F_n(f_1(x_1),\ldots,f_n(x_n)) \mid (x_1,\ldots,x_n) \in \underline{S}_n(\theta_n,\alpha_n) \}$$

$$= \max\{ h_n(F_{n-1},f_n) \mid (x_1,\ldots,x_n) \in \underline{S}_n(\theta_n,\alpha_n) \} \ , \qquad (8.13)$$

$$\bar{S}_{mn}(\theta_{mn},\alpha_{mn}) \equiv \{ (x_1,\ldots,x_n) \ ;$$

$$\int\cdots\int_{\bigcup_{\delta_{mn}\in D_{mn}(\theta_{mn})} V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1})}$$

$$\times \, d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1})$$

$$\times \int_{\bigcup_{\delta_{mn}\in D_{mn}(\theta_{mn})} v_{mn}(x_n,\theta_{mn})} d\phi_n(a_n) \geqq \alpha_{mn}$$

$$\text{and} \quad x_k \in X_k \ , \quad k=1,\ldots,n \} \ , \qquad (8.14)$$

$$\bar{S}_n(\theta_n,\alpha_n) \equiv \mathop{\times}_{m=1}^{M} \bar{S}_{mn}(\theta_{mn},\alpha_{mn}) \qquad (8.15)$$

and

$$\bar{P}_n(\theta_n,\alpha_n) \equiv \max\{ F_n(f_1(x_1),\ldots,f_n(x_n) \mid (x_1,\ldots,x_n) \in \bar{S}_n(\theta_n,\alpha_n) \}$$

$$= \max\{ h_n(F_{n-1},f_n) \mid (x_1,\ldots,x_n) \in \bar{S}_n(\theta_n,\alpha_n) \} \ . \qquad (8.16)$$

In (8.9) and (8.10), the suprema are assumed to be finite. Furthermore, define for $m=1,\ldots,M$ and $n=1,\ldots,N$,

$$s_{mn}(\delta_{mn},\beta_{mn}) \equiv \{ x_n \; ; \; \Pr\{ g_{mn}(a_n,x_n) \leqq \delta_{mn} \} \geqq \beta_{mn} \quad \text{and} \quad x_n \in X_n \}$$

$$= \{ x_n \; ; \; \int_{v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \beta_{mn} \quad \text{and} \quad x_n \in X_n \}, \tag{8.17}$$

$$s_n(\delta_n,\beta_n) \equiv \bigcap_{m=1}^{M} s_{mn}(\delta_{mn},\beta_{mn}) \;, \tag{8.18}$$

and

$$p_n(\delta_n,\beta_n) \equiv \max\{ f_n(x_n) \mid x_n \in s_n(\delta_n,\beta_n) \} \;, \tag{8.19}$$

where $\delta_n = (\delta_{1n},\ldots,\delta_{Mn})$ and $\beta_n = (\beta_{1n},\ldots,\beta_{Mn})$.

8.3 Main Results

The following theorem gives a sufficient condition for the principal problem to be decomposed into subproblems and to be solved approximately by dynamic programming.

Theorem 8.1. Assume that the objective function F and the constraint functions G_m , $m=1,\dots,M$, in the principal problem (8.1) are strongly decomposable with separating functions h_n , $n=1,\dots,N$, and h_{mn} , $m=1,\dots,M$, $n=1,\dots,N$, respectively. Then the following relations hold for $n=2,\dots,N$:

$$\underline{P}_n(\theta_n,\alpha_n) = \max\{ h_n(\underline{P}_{n-1}(\theta_{n-1},\alpha_{n-1}),p_n(\delta_n,\beta_n)) \mid \delta_n \in D_n(\theta_n) \text{ and } \alpha_n \leqq \beta_n \leqq 1 \} \tag{8.20}$$

and

$$\bar{P}_n(\theta_n,\alpha_n) = \max\{ h_n(\bar{P}_{n-1}(\bar{\theta}_{n-1},\alpha_{n-1}),p_n(\bar{\delta}_n,\beta_n)) \mid \alpha_n \leqq \beta_n \leqq 1 \} , \tag{8.21}$$

where

$$\alpha_{n-1} = (\alpha_{1n-1},\dots,\alpha_{Mn-1}) = (\alpha_{1n}/\beta_{1n},\dots,\alpha_{Mn}/\beta_{Mn}) \tag{8.22}$$

and $\alpha_n \leqq \beta_n \leqq 1$ means

$$\alpha_{mn} \leqq \beta_{mn} \leqq 1 , \quad m=1,\dots,M .$$

Moreover, for $n=2,\dots,N$,

$$\underline{P}_n(\theta_n,\alpha_n) \leqq P_n(\theta_n,\alpha_n) \leqq \bar{P}_n(\theta_n,\alpha_n) . \tag{8.23}$$

Proof. From (8.11), (8.12), (8.17), and (8.18), for $n = 2,\dots,N$,

$$\underline{S}_{mn}(\theta_{mn},\alpha_{mn})$$

$$= \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} \ \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [\{ (x_1,\dots,x_{n-1}) ;$$

$$\int\cdots\int_{V_{mn-1}(x_1,\dots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\cdots d\phi_{n-1}(a_{n-1}) \geqq$$

$$\alpha_{mn}/\beta_{mn} \text{ and } x_k \in X_k ,\ k = 1,\dots,n-1 \} \times \{ x_n ;$$

$$\int_{V_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \beta_{mn} \text{ and } x_n \in X_n \}]$$

$$= \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} \ \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [S_{mn-1}(\theta_{mn-1},\alpha_{mn-1}) \times s_{mn}(\delta_{mn},\beta_{mn})]$$

and

$$\underline{S}_n(\theta_n,\alpha_n)$$

$$= \bigcap_{m=1}^{M} \{ \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} \ \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [S_{mn-1}(\theta_{mn-1},\alpha_{mn-1})$$

$$\times s_{mn}(\delta_{mn},\beta_{mn})] \}$$

$$= \bigcup_{\delta_n \in D_n(\theta_n)} \ \bigcup_{\alpha_n \leqq \beta_n \leqq 1} [\bigcap_{m=1}^{M} S_{mn-1}(\theta_{mn-1},\alpha_{mn-1}) \times \bigcap_{m=1}^{M} s_{mn}(\delta_{mn},\beta_{mn})]$$

$$= \bigcup_{\delta_n \in D_n(\theta_n)} \ \bigcup_{\alpha_n \leqq \beta_n \leqq 1} [S_{n-1}(\theta_{n-1},\alpha_{n-1}) \times s_n(\delta_n,\beta_n)] , \qquad (8.24)$$

where α_{n-1} is given by (8.22). Hence, by (8.5), (8.13), (8.19), (8.24) and the monotonicity of the separating functions h_n of F, we have for $n=2,\dots,N$,

$$\underline{P}_n(\theta_n,\alpha_n)$$

$$= \max\{ h_n(F_{n-1}(f_1(x_1),\dots,f_{n-1}(x_{n-1})),f_n(x_n)) \mid (x_1,\dots,x_n) \in \bigcup_{\delta_n \in D_n(\theta_n)} \bigcup_{\alpha_n \leqq \beta_n \leqq 1} [S_{n-1}(\theta_{n-1},\alpha_{n-1}) \times s_n(\delta_n,\beta_n)]\}$$

$$= \max\{ h_n(\max\{F_{n-1}(f_1(x_1),\dots,f_{n-1}(x_{n-1})) \mid (x_1,\dots,x_{n-1}) \in S_{n-1}(\theta_{n-1},\alpha_{n-1})\}, \max\{f_n(x_n) \mid x_n \in s_n(\delta_n,\beta_n)\}) \mid \delta_n \in D_n(\theta_n) \text{ and } \alpha_n \leqq \beta_n \leqq 1\}$$

$$= \max\{ h_n(P_{n-1}(\theta_{n-1},\alpha_{n-1}),p_n(\delta_n,\beta_n)) \mid \delta_n \in D_n(\theta_n) \text{ and } \alpha_n \leqq \beta_n \leqq 1\}.$$

Next, we prove (8.21). From (8.14), for $n=2,\dots,N$,

$$\bar{S}_{mn}(\theta_{mn},\alpha_{mn})$$

$$= \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [\{(x_1,\dots,x_{n-1});\ \int\!\cdots\!\int_{\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} V_{mn-1}(x_1,\dots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\dots d\phi_{n-1}(a_{n-1}) \geqq \alpha_{mn}/\beta_{mn} \text{ and } x_k \in X_k,\ k=1,\dots,n-1\} \times \{x_n;\ \int_{\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \beta_{mn} \text{ and } x_n \in X_n\}].$$

It is clear by the definition of V_{mn-1} that $\theta_{mn-1} \geqq \theta'_{mn-1}$ implies

$$V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \subset V_{mn-1}(x_1,\ldots,x_{n-1},\theta'_{mn-1}) \ . \qquad (8.25)$$

Therefore, we have

$$\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) = V_{mn-1}(x_1,\ldots,x_{n-1},\bar{\theta}_{mn-1}) \ ,$$

where $\bar{\theta}_{mn-1}$ is defined by (8.9). Similarly, since by (8.7), $\delta_{mn} \geqq \delta'_{mn}$ implies

$$v_{mn}(x_n,\delta_{mn}) \subset v_{mn}(x_n,\delta'_{mn}) \ , \qquad (8.26)$$

we have

$$\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} v_{mn}(x_n,\delta_{mn}) \subset v_{mn}(x_n,\bar{\delta}_{mn}) \ ,$$

where $\bar{\delta}_{mn}$ is defined by (8.10). Note that $\bar{\theta}_{mn-1}$ and $\bar{\delta}_{mn}$ are dependent on θ_{mn} . Hence, from (8.14), (8.15), (8.17), and (8.18), it follows that for $n = 2,\ldots,N$,

$$\bar{S}_{mn}(\theta_{mn},\alpha_{mn})$$

$$= \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [\ (x_1,\ldots,x_{n-1}) \ ;$$

$$\int \cdots \int_{V_{mn-1}(x_1,\ldots,x_{n-1},\bar{\theta}_{mn-1})} d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1}) \geqq \alpha_{mn}/\beta_{mn}$$

$$\text{and} \quad x_k \in X_k \ , \ k = 1,\ldots,n-1 \ \} \times \{ \ x_n \ ; \int_{v_{mn}(x_n,\bar{\delta}_{mn})} d\phi_n(a_n)$$

$$\geqq \beta_{mn} \quad \text{and} \quad x_n \in X_n \ \} \ ]$$

$$= \bigcup_{\alpha_{mn} \leqq \beta_{mn} \leqq 1} [\ S_{mn-1}(\bar{\theta}_{mn-1},\alpha_{mn-1}) \times s_{mn}(\bar{\delta}_{mn},\beta_{mn}) \ ]$$

and

$$\bar{S}_{mn}(\theta_n,\alpha_n) = \bigcap_{m=1}^{M} \{ \bigcup_{\alpha_{mn} \leq \beta_{mn} \leq 1} [S_{mn-1}(\bar{\theta}_{mn-1},\alpha_{mn-1}) \times s_{mn}(\bar{\delta}_{mn},\beta_{mn})] \}$$

$$= \bigcup_{\alpha_n \leq \beta_n \leq 1} [\bigcap_{m=1}^{M} S_{mn-1}(\bar{\theta}_{mn-1},\alpha_{mn-1}) \times \bigcap_{m=1}^{M} s_{mn}(\bar{\delta}_{mn},\beta_{mn})]$$

$$= \bigcup_{\alpha_n \leq \beta_n \leq 1} [S_{n-1}(\bar{\theta}_{n-1},\alpha_{n-1}) \times s_n(\bar{\delta}_n,\beta_n)] .$$

Consequently, by (8.16) and (8.19), and the monotonicity of the separating functions h_n of F, we have for $n = 2,\ldots,N$,

$$\bar{P}_n(\theta_n,\alpha_n)$$

$$= \max \{ h_n(F_{n-1}(f_1(x_1),\ldots,f_{n-1}(x_{n-1})),f_n(x_n)) \mid (x_1,\ldots,x_n) \in \bigcup_{\alpha_n \leq \beta_n \leq 1} [S_{n-1}(\bar{\theta}_{n-1},\alpha_{n-1}) \times s_n(\bar{\delta}_n,\beta_n)] \}$$

$$= \max \{ h_n(\max \{ F_{n-1}(f_1(x_1),\ldots,f_{n-1}(x_{n-1})) \mid (x_1,\ldots,x_{n-1}) \in S_{n-1}(\bar{\theta}_{n-1},\alpha_{n-1}) \} , \max \{ f_n(x_n) \mid x_n \in s_n(\bar{\delta}_n,\beta_n) \mid \alpha_n \leq \beta_n \leq 1 \}$$

$$= \max \{ h_n(P_{n-1}(\bar{\theta}_{n-1},\alpha_{n-1}),p_n(\bar{\delta}_n,\beta_n)) \mid \alpha_n \leq \beta_n \leq 1 \} .$$

Finally, we prove (8.23). In view of (8.5), (8.13) and (8.16), it suffices to show that, for $n = 2,\ldots,N$,

$$\underline{S}_n(\theta_n,\alpha_n) \subset S_n(\theta_n,\alpha_n) \subset \bar{S}_n(\theta_n,\alpha_n) .$$

Since for $m = 1,\ldots,M$ and $n = 2,\ldots,N$,

$$V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})$$

$$\subset \bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [\, V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})\,] \quad ,$$

we have

$$\int \cdots \int_{V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1}) \cdot \int_{v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n)$$

$$\leqq \int \cdots \int_{\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})]} d\phi_1(a_1)\ldots d\phi_n(a_n) \; .$$

This implies that

$$\{ (x_1,\ldots,x_n) \; ; \int \cdots \int_{V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1})$$

$$\times \int_{v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k \;,\; k=1,\ldots,n \}$$

$$\subset \{ (x_1,\ldots,x_n) \; ; \int \cdots \int_{\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})]}$$

$$\times \, d\phi_1(a_1)\ldots d\phi_n(a_n) \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k \;,\; k=1,\ldots,n \} \; .$$

Therefore, from (8.8) and (8.11),

$$\underline{S}_{mn}(\theta_{mn},\alpha_{mn}) \subset S_{mn}(\theta_{mn},\alpha_{mn})$$

and hence

$$\underline{S}_n(\theta_n,\alpha_n) \subset S_n(\theta_n,\alpha_n) \; .$$

Similarly, from (8.9), (8.10), (8.25), and (8.26), it follows that for $m=1,\ldots,M$ and $n=2,\ldots,N$,

$$\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})]$$

$$\subset \ V_{mn-1}(x_1,\ldots,x_{n-1},\bar{\theta}_{mn-1}) \times v_{mn}(x_n,\bar{\delta}_{mn})$$

and

$$\{ (x_1,\ldots,x_n) \ ; \int \cdots \int_{\bigcup_{\delta_{mn} \in D_{mn}(\theta_{mn})} [V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1}) \times v_{mn}(x_n,\delta_{mn})]}$$

$$\times \ d\phi_1(a_1)\ldots d\phi_n(a_n) \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k \ , \ k = 1,\ldots,n \ \}$$

$$\subset \ \{ (x_1,\ldots,x_n) \ ; \int \cdots \int_{V_{mn-1}(x_1,\ldots,x_{n-1},\theta_{mn-1})} d\phi_1(a_1)\ldots d\phi_{n-1}(a_{n-1})$$

$$\cdot \int_{v_{mn}(x_n,\delta_{mn})} d\phi_n(a_n) \geqq \alpha_{mn} \quad \text{and} \quad x_k \in X_k \ , \ k = 1,\ldots,n \ \} .$$

Therefore, by (8.8) and (8.14), we have

$$S_{mn}(\theta_{mn},\alpha_{mn}) \subset \bar{S}_{mn}(\theta_{mn},\alpha_{mn})$$

and hence

$$S_n(\theta_n,\alpha_n) \subset \bar{S}_n(\theta_n,\alpha_n) \ .$$

The proof is completed. □

According to this theorem, a lower bound of the maximum of the principal problem (8.1) is obtained by the following algorithm:

<u>Algorithm</u>.

Step 1: Solve for every θ_1 and $\alpha \leqq \alpha_1 \leqq 1$,

$$P_1(\theta_1,\alpha_1) = p_1(\theta_1,\alpha_1)$$

$$= \max \{ f_1(x_1) \mid x_1 \in s_1(\theta_1,\alpha_1) \}$$

$$= f_1(x_1^*(\theta_1,\alpha_1)) .$$

Then let $n = 2$ and go to Step 2.

Step 2: Solve for all δ_n and $\alpha \leqq \beta_n \leqq 1$,

$$p_n(\delta_n,\beta_n) = \max \{ f_n(x_n) \mid x_n \in s_n(\delta_n,\beta_n) \}$$

$$= f_n(x_n^*(\delta_n,\beta_n)) .$$

Step 3: Calculate, for all θ_n and $\alpha \leqq \alpha_n \leqq 1$, $\underline{P}_n(\theta_n,\alpha_n)$ by (8.20) where $\underline{P}_{n-1}(\theta_{n-1},\alpha_{n-1})$ is used in place of $P_{n-1}(\theta_{n-1},\alpha_{n-1})$, which was already obtained as well as $p_n(\delta_n,\beta_n)$. Then put $\theta^*_{n-1}(\theta_n,\alpha_n)$, $\alpha^*_{n-1}(\theta_n,\alpha_n)$ and $\delta^*_n(\theta_n,\alpha_n)$, $\beta^*_n(\theta_n,\alpha_n)$ which achieve the maximum in (8.20).

Step 4: Set $n = n+1$ and go to Step 5 if $n > N$; otherwise go to Step 2.

Step 5: A lower bound of the maximal value of the principal problem is given by $\underline{P}_N(0,\alpha)$.

Step 6: Put $\theta^{**}_{N-1} = \theta^*_{N-1}(0,\alpha)$, $\alpha^{**}_{N-1} = \alpha^*_{N-1}(0,\alpha)$, $\delta^{**}_N = \delta^*_N(0,\alpha)$, $\beta^{**}_N = \beta^*_N(0,\alpha)$ and calculate $x^*_N(\delta^{**}_N,\beta^{**}_N)$. Then let $n = N-1$ and go to Step 7.

Step 7: Put $\theta^{**}_{n-1} = \theta^*_{n-1}(\theta^{**}_n,\alpha^{**}_n)$, $\alpha^{**}_{n-1} = \alpha^*_{n-1}(\theta^{**}_n,\alpha^{**}_n)$, $\delta^{**}_n = \delta^*_n(\theta^{**}_n,\alpha^{**}_n)$, $\beta^{**}_n = \beta^*_n(\theta^{**}_n,\alpha^{**}_n)$ and calculate $x^*_n(\delta^{**}_n,\beta^{**}_n)$.

Step 8: Set $n = n-1$ and go to Step 9 if $n < 2$; otherwise go to Step 7.

Step 9: Calculate $x_1^*(\theta_1^{**},\alpha_1^{**})$ and determine the solution as

$$\underline{x}^* = (x_1^*(\theta_1^{**},\alpha_1^{**}), x_2^*(\theta_2^{**},\alpha_2^{**}), \ldots, x_N^*(\theta_N^{**},\alpha_N^{**})) .$$

On the other hand, in a way similar to the above, using (8.21) we can obtain an upper bound of the maximal value, say $\bar{P}_N(0,\alpha)$, and the corresponding solution $\bar{x}^*$ of the principal problem (8.1). It may be expected that, if the difference between $\underline{P}_N(0,\alpha)$ and $\bar{P}_N(0,\alpha)$ is sufficiently small, then $\underline{P}_N(0,\alpha)$ and $\bar{P}_N(0,\alpha)$ can be regarded as good approximate maximal values of the principal problem. Furthermore, in such cases, $\underline{x}^*$ and $\bar{x}^*$ are expected to be sufficiently good approximations of the optimal solution of the principal problem (8.1).

8.4 Example

In this section, we consider the following problem:

$$\begin{aligned}
&\text{maximize} && \sum_{n=1}^{N} f_n(x_n) \\
&\text{subject to} && \Pr\{ \sum_{n=1}^{N} a_n x_n \leqq b \} \geqq \alpha && (8.27) \\
&\text{and} && x_n \geqq 0 , \qquad n = 1,\ldots,N ,
\end{aligned}$$

where x_n , $n = 1,\ldots,N$, are real numbers and a_n , $n = 1,\ldots,N$, are random variables which are distributed uniformly in the interval $[0,10]$. Assume that the functions f_n are nondecreasing for $x_n \geqq 0$. It follws from (8.17) that, for $n = 1,\ldots,N$,

$$\begin{aligned}
s_n(\delta_n,\beta_n) &= \{ x_n ; \Pr(a_n x_n \leqq \delta_n) \geqq \beta_n \text{ and } x_n \geqq 0 \} \\
&= \{ x_n ; 0 \leqq x_n \leqq \delta_n/10\beta_n \} ,
\end{aligned}$$

where $0 \leqq \delta_n \leqq b$ and $\alpha \leqq \beta_n \leqq 1$. By the monotonicity of f_n , nth subproblems can be solved as

$$\begin{aligned}
p_n(\delta_n,\beta_n) &= \max \{ f_n(x_n) \mid 0 \leqq x_n \leqq \delta_n/10\beta_n \} \\
&= f_n(\delta_n/10\beta_n) . && (8.28)
\end{aligned}$$

Consequently, by (8.20) and (8.21), the following recursive relations yield an approximate solution of $P = P_N(b,\alpha)$, the principal problem (8.27):

$$P_1(\theta_1,\alpha_1) = p_1(\theta_1,\alpha_1) = f_1(\theta_1/10\alpha_1)$$

and for $n = 2,\ldots,N$,

$$\underline{P}_n(\theta_n,\alpha_n) = \max \{ \underline{P}_{n-1}(\theta_n - \delta_n, \alpha_n/\beta_n) + p_n(\delta_n,\beta_n) \mid$$

$$0 \leqq \delta_n \leqq \theta_n \quad \text{and} \quad \alpha_n \leqq \beta_n \leqq 1 \} , \tag{8.29}$$

and

$$\bar{P}_n(\theta_n,\alpha_n) = \max \{ \bar{P}_{n-1}(\theta_n,\alpha_n/\beta_n) + p_n(\theta_n,\beta_n) \mid \alpha_n \leqq \beta_n \leqq 1 \} , \tag{8.30}$$

where P_n , $n=2,\ldots,N$, are approximated by $\underline{P}_n$ and $\bar{P}_n$, respectively.

A numerical example of the type (8.27) is given in which $N = 2$, $f_1(x_1) = 5x_1$, $f_2(x_2) = x^2$, $b = 1$ and $\alpha = 0.8$. Then by (8.28), (8.29), and (8.30), we obtain

$$\underline{P} = \underline{P}_2(1,0.8) = \max \{ 5(1-\delta)/10(0.8/\beta) + (\delta/10\beta)^2 \mid$$

$$0 \leqq \delta \leqq 1 \quad \text{and} \quad 0.8 \leqq \beta \leqq 1 \}$$

$$= 0.625 \qquad (\delta = 0 , \beta = 1) ,$$

and

$$\bar{P} = \bar{P}_2(1,0.8) = \max \{ 5 \times 1/10(0.8/\beta) + (10/10\beta)^2 \mid 0.8 \leqq \beta \leqq 1 \}$$

$$= 0.626 \qquad (\beta = 1) .$$

The solutions of $\underline{P}$ and $\bar{P}$ are $(\underline{x}_1^*,\underline{x}_2^*) = (0.125,0)$ and $(\bar{x}_1^*,\bar{x}_2^*) = (0.125,0.1)$, respectively. Consequently, the optimal value P of the principal problem (8.27) satisfies

$$0.625 \leqq P \leqq 0.626 .$$

8.5 Conclusion

In this chapter, a method of solving a chance-constrained programming problem with a nonlinear objective function and non-linear constraint functions has been presented. It should be noted that this method does not require random variables to be specifically distributed, while most other methods are concerned with particular distributions like the normal distribution or the chi-square distribution [Sl]. Because of the inherent complexity, however, it is inevitable that this method provides only approximate optimal solutions. Although it depends upon the original problem whether such approximation is allowable, this method gives good approximation for some problems which have never been solved.

CHAPTER 9

DECOMPOSITION OF MULTIPLE CRITERIA MATHEMATICAL PROGRAMMING PROBLEMS BY DYNAMIC PROGRAMMING

In this chapter, we give a method of generating efficient and properly efficient solutions of a multiple criteria mathematical programming problem. The method is based on the principle of optimality of dynamic programming in the similar way as in the last two chapters. Assuming the separability and monotonicity of the problem, we derive generalized recursive functional equations of dynamic programming. Moreover, we discuss some computational procedures which reduce multiple criteria problems into scalar criterion problems.

9.1 Introduction

Recently, multiple criteria optimization problems have been studied extensively from various viewpoints [C5][K3][Z3]. Unfortunately, unlike problems with a scalar criterion, it is not a trivial matter to define the concept of optimality since generally multiple criteria problems do not have feasible solutions which simultaneously maximize all criteria. At present, the most approved concept of optimality for multiple criteria might be that of efficiency, or Pareto optimality [K3][K6]. An efficient solution is a feasible solution such that value of any criterion cannot be increased without decreasing the value of at least one other. More recently, properly efficient solutions have been characterized by Geoffrion [G2] in order to exclude certain efficient solutions which are unusual in some sense. Since efficient or properly efficient solutions of a multiple criteria problem are, in general, not unique and any efficient solution is as preferable as any other efficient solutions as long as we do not take account of a certain additional criterion such as a utility function, it seems important to find a set of all or possibly some efficient solutions. To this end, there have been a number of techniques devised to obtain a set of efficient solutions.

In this chapter, we formulate a method of generating efficient and properly efficient solutions of mathematical programming problems with multiple objective functions which possess certain structure. The method is based on the principle of optimality of dynamic programming [B2][N1] and is formally a natural extension of the results for single objective mathematical programming problems presented in Chapter 7. However, the present method takes the characteristics of the multiple criteria problems into account, and thus has little in common with the proofs of the theorems of the previous chapter.

This chapter is organized as follws: In Section 9.2, some definitions are given and multiple criteria subproblems are formulated. Moreover, several notions used in the subsequent sections are also cited. In Section 9.3, relationships between the sets of efficient or properly efficient solutions of subproblems are examined, and a recursive formula of dynamic programming is derived. In Section 9.4, some computational aspects for this approach are discussed.

9.2 The problem and Definitions

The multiple criteria methematical programming problem considered in this chapter is the following:

$$\begin{array}{llll} \text{maximize} & F_\ell(f_{\ell 1}(x_1),\dots,f_{\ell N}(x_N)) , & \ell=1,\dots,L \ (L \geqq 2) , & \\ \text{subject to} & G_m(g_{m1}(x_1),\dots,g_{mN}(x_N)) \leqq 0 , & m=1,\dots,M , & (9.1) \\ \text{and} & x_n \in X_n , & n=1,\dots,N , & \end{array}$$

where for each $n=1,\dots,N$, X_n is a subset of R^{k_n} and x_n is a k_n-vector, and the objective functions F_ℓ, $\ell=1,\dots,L$, and the constraint functions G_m, $m=1,\dots,M$, are real valued functions on R^N, and $f_{\ell n}$ and g_{mn}, $\ell=1,\dots,L$, $m=1,\dots,M$, $n=1,\dots,N$, are real valued functions on X_n. In the following, we shall sometimes use the vector notation

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_L \end{bmatrix}, \quad G = \begin{bmatrix} G_1 \\ \vdots \\ G_M \end{bmatrix}, \quad f_n = \begin{bmatrix} f_{1n} \\ \vdots \\ f_{Ln} \end{bmatrix}, \text{ and } g_n = \begin{bmatrix} g_{1n} \\ \vdots \\ g_{Mn} \end{bmatrix},$$

for $n=1,\dots,N$.

Separability and monotonicity of functions have been utilized for deriving the recursive formulas of dynamic programming in the previous chapters. We give here definitions of those properties for problem (9.1).

The objective function F_ℓ is said to be *separable* if there exist functions F_ℓ^n, $n=1,\dots,N$, defined on R^n and functions ϕ_ℓ^n, $n=2,\dots,N$, defined on R^2 satisfying, for $n=2,\dots,N$,

$$F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n}(x_n)) = \phi_\ell^n(F_\ell^{n-1}(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1})),f_{\ell n}(x_n)) \tag{9.2}$$

and

$$F_\ell^N(f_{\ell 1}(x_1),\ldots,f_{\ell N}(x_N)) = F_\ell(f_{\ell 1}(x_1),\ldots,f_{\ell N}(x_N)) \; .$$

Similarly, the constraint function G_m is separable if for some functions G_m^n, $n=1,\ldots,N$, on R^n and ψ_m^n, $n=2,\ldots,N$, on R^2, it holds that for $n=2,\ldots,N$,

$$G_m^n(g_{m1}(x_1),\ldots,g_{mn}(x_n)) = \psi_m^n(G_m^{n-1}(g_{m1}(x_1),\ldots,g_{mn-1}(x_{n-1})),g_{mn}(x_n)) \tag{9.3}$$

and

$$G_m^N(g_{m1}(x_1),\ldots,g_{mN}(x_N)) = G_m(g_{m1}(x_1),\ldots,g_{mN}(x_N)) \; .$$

If all objective and constraint functions are separable, we say that problem (9.1) is *separable*. Furthermore, the separation of problem (9.1) is said to be *monotone* if all functions ϕ_ℓ^n and ψ_m^n are strictly increasing with respect to the first argument for each fixed second argument. Specifically, for each $y \in R$,

$$\phi_\ell^n(z,y) > \phi_\ell^n(z',y) \quad \text{if and only if} \quad z > z'$$

and

$$\psi_m^n(z,y) > \psi_m^n(z',y) \quad \text{if and only if} \quad z > z'$$

for every $\ell=1,\ldots,L$, $m=1,\ldots,M$, and $n=2,\ldots,N$.

Assuming the separability of problem (9.1), we may define an *nth subproblem*, which we denote $P_n[z]$, for each $n=1,\ldots,N$, and each $z=(z_1,\ldots,z_m) \in R^m$ as follows:

$$\begin{aligned} &\text{maximize} && F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) \,, && \ell = 1,\ldots,L \,, \\ &\text{subject to} && G_m^n(g_{m1}(x_1),\ldots,g_{mn}(x_n)) \leqq z_m \,, && m = 1,\ldots,M \,, \\ &\text{and} && x_k \in X_k \,, && k = 1,\ldots,n \,. \end{aligned}$$

It is clear that, when $n = N$ and $z = 0$, the above problem coincides with problem (9.1). In the following, we shall often use the convention $F^n = (F_1^n,\ldots,F_L^n)$, $G = (G_1^n,\ldots,G_M^n)$, ..., etc. The nth subproblem $P_n[z]$ for each $z \in R^m$ is again a multiple criteria mathematical programming problem.

Now we characterize efficient solutions of problem $P_n[z]$. Let $(x_1^o,\ldots,x_n^o)$ be a feasible solution to problem $P_n[z]$. It is called *efficient*, or *Pareto optimal*, if there exists no feasible solution $(x_1,\ldots,x_n)$ such that

$$F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) \geqq F_\ell^n(f_{\ell 1}(x_1^o),\ldots,f_{\ell n}(x_n^o))$$

for all ℓ and

$$F_k^n(f_{k1}(x_1),\ldots,f_{kn}(x_n)) > F_k^n(f_{k1}(x_1^o),\ldots,f_{kn}(x_n^o))$$

for at least one index k. Equivalently, $(x_1^o,\ldots,x_n^o)$ is efficient if

$$F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) \geqq F_\ell^n(f_{\ell 1}(x_1^o),\ldots,f_{\ell n}(x_n^o))$$

for all ℓ and feasible $(x_1,\ldots,x_n)$ implies

$$F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) = F_\ell^n(f_{\ell 1}(x_1^o),\ldots,f_{\ell n}(x_n^o))$$

for all ℓ .

Properly efficient solutions may be characterized as those particular efficient solutions for each of which ratio of marginal gain in some criterion to marginal loss in another criterion is bounded from above. More precisely, we can state as follows: Let $(x_1^o,\dots,x_n^o)$ be an efficient solution to problem $P_n[z]$. It is called *properly efficient* if there exists a scalar $\mu > 0$ such that, for each k and each feasible solution $(x_1,\dots,x_n)$ satisfying

$$F_k^n(f_{k1}(x_1),\dots,f_{kn}(x_n)) > F_k^n(f_{k1}(x_1^o),\dots,f_{kn}(x_n^o)) ,$$

there exists at least one another index ℓ such that

$$F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n}(x_n)) < F_\ell^n(f_{\ell 1}(x_1^o),\dots,f_{\ell n}(x_n^o))$$

and

$$\frac{F_k^n(f_{k1}(x_1),\dots,f_{kn}(x_n)) - F_k^n(f_{k1}(x_1^o),\dots,f_{kn}(x_n^o))}{F_\ell^n(f_{\ell 1}(x_1^o),\dots,f_{\ell n}(x_n^o)) - F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n}(x_n))} \leqq \mu .$$

For each $n=1,\dots,N$ and $z \in R^m$, the set of all efficient solutions of problem $P_n[z]$ is denoted by $E_n[z]$ and the set of all properly efficient solutions is denoted by $PE_n[z]$. Therefore, to solve problem (9.1) may be interpreted as finding the set $E_N[0]$ or $PE_N[0]$.

For a given set Y in R^L, a point $y^o \in Y$ is called a *maximal point* of Y if there exists no point $y \in Y$ such that $y_\ell \geqq y_\ell^o$ for all ℓ and $y_k > y_k^o$ for at least one k. The set of all maximal points of Y is denoted by Maximal Y.

It is clear from the definition that a feasible solution $(x_1^o,\ldots,x_n^o)$ of problem $P_n[z]$ is efficient if and only if the corresponding point $F^n(f_1(x_1^o),\ldots,f_n(x_n^o))$ in R^L is a maximal point in the image of the feasible set of problem $P_n[z]$ under the mapping F^n (see [L4]).

Before proceeding to the next section, we define for $n=2,\ldots,N$ functions $z^{n-1}(\cdot,\cdot) = (z_1^{n-1}(\cdot,\cdot),\ldots,z_M^{n-1}(\cdot,\cdot)) : R^{k_n} \times R^M \to R^M$ as follows: Assuming the monotonicity of ψ_m^n, let z_m^{n-1} be defined by

$$z_m^{n-1}(x_n,z) = \sup \{ \xi \in R ; \psi_m^n(\xi,g_{mn}(x_n)) \leqq z_m \} , \quad m=1,\ldots,M . \tag{9.4}$$

These functions will play a crucial role in the subsequent discussions.

9.3 Main Results

In this section, we shall clarify the relationship between the set of (properly) efficient solutions of the nth subproblem and that of the (n-1)th subproblem. Let us focus our attention to a particular nth subproblem, where n is any integer such that $2 \leqq n \leqq N$. The results of this section are based on the following assumptions.

Assumption 9.1. Problem (9.1) is separable and the separation is monotone.

Assumption 9.2. For every n, X_n is compact and the functions $F^n_\ell(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n))$, $\ell = 1,\ldots,L$, and $G^n_m(g_{m1}(x_1),\ldots,g_{mn}(x_n))$, $m = 1,\ldots,M$, are continuous with respect to $(x_1,\ldots,x_n)$.

The first assumption is essential in dealing with the problem by dynamic programming, as we have seen in the previous chapters. The second assumption assures the property that for any nonefficient solution there exists some efficient solution dominating it.

Now we state our first result.

Theorem 9.1. Suppose that Assumptions 9.1 and 9.2 hold. Let $(x^o_1,\ldots,x^o_n)$ be any efficient solution of problem $P_n[z]$. Then $(x^o_1,\ldots,x^o_{n-1})$ is an efficient solution of problem $P_{n-1}[z^{n-1}(x^o_n,z)]$.

Proof. Assume that $(x_1^o,\dots,x_{n-1}^o) \notin E_{n-1}[z^{n-1}(x_n^o,z)]$. Then there must exist some $(x_1,\dots,x_{n-1})$ satisfying

$$G_m^{n-1}(g_{m1}(x_1),\dots,g_{mn-1}(x_{n-1}) \leqq z_{mn-1}(x_n^o,z) \ , \ m=1,\dots,M \ , \tag{9.5}$$

and

$$\left.\begin{aligned} &F_\ell^{n-1}(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1})) \\ &\qquad \geqq F_\ell^{n-1}(f_{\ell 1}(x_1^o),\dots,f_{\ell n-1}(x_{n-1}^o)) \quad \text{for all } \ell \\ &F_k^{n-1}(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1})) \\ &\qquad > F_k^{n-1}(f_{k1}(x_1^o),\dots,f_{kn-1}(x_{n-1}^o)) \quad \text{for some } k. \end{aligned}\right\} \tag{9.6}$$

By (9.3), (9.4), (9.5) and the monotonicity of ψ_m^n, we have

$$G_m^n(g_{m1}(x_1),\dots,g_{mn-1}(x_{n-1}),g_{mn}(x_n^o)) \leqq z_m \ ,$$

i.e., $(x_1,\dots,x_{n-1},x_n^o)$ is feasible to problem $P_n[z]$. Moreover, it follows from (9.2), (9.6) and the monotonicity of ϕ_ℓ^n that

$$\begin{aligned} &F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1}),f_{\ell n}(x_n^o)) \\ &\qquad \geqq F_\ell^n(f_{\ell 1}(x_1^o),\dots,f_{\ell n}(x_n^o)) \quad \text{for all } \ell \ , \end{aligned}$$

and

$$\begin{aligned} &F_k^n(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1}),f_{kn}(x_n^o)) \\ &\qquad > F_k^n(f_{k1}(x_1^o),\dots,f_{kn}(x_n^o)) \quad \text{for some } k \ . \end{aligned}$$

This shows that $(x_1^o,\dots,x_n^o) \notin E_n[z]$, which is a contradiction. The proof is complete. □

Note that the converse of the above result does not necessarily hold. Namely, even if $(x_1^o,\ldots,x_{n-1}^o) \in E_{n-1}[z]$, it is generally not true that $(x_1^o,\ldots,x_{n-1}^o,x_n) \in E_n[\psi^n(z,g_n(x_n))]$ for all, or even some, $x_n \in X_n$. However, we can prove the following relationship.

Theorem 9.2. Suppose that Assumptions 9.1 and 9.2 hold. Let $(x_1^o,\ldots,x_n^o)$ be any efficient solution of the following multiple criteria problem:

$$\text{maximize} \quad F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) \;, \quad \ell = 1,\ldots,L \;,$$
$$\text{subject to} \qquad (9.7)$$
$$(x_1,\ldots,x_n) \in \bigcup_{x_n \in X_n} E_{n-1}[z^{n-1}(x_n,z)] \times \{x_n\} \;.$$

Then the $(x_1^o,\ldots,x_n^o)$ constitutes an efficient solution of problem $P_n[z]$.

Proof. Obviously every feasible solution to (9.7) is also feasible to $P_n[z]$. Suppose that there exists some $(x_1,\ldots,x_n)$ feasible to problem $P_n[z]$ such that

$$\left.\begin{aligned} &F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n}(x_n)) \\ &\qquad \geqq F_\ell^n(f_{\ell 1}(x_1^o),\ldots,f_{\ell n}(x_n^o)) \quad \text{for all} \quad \ell \;, \\ &F_k^n(f_{k1}(x_1),\ldots,f_{kn}(x_n)) \\ &\qquad > F_k^n(f_{k1}(x_1^o),\ldots,f_{kn}(x_n^o)) \quad \text{for some} \quad k \;. \end{aligned}\right\} \qquad (9.8)$$

Let us now consider problem $P_{n-1}[z^{n-1}(x_n,z)]$. Clearly the point $(x_1,\ldots,x_{n-1})$ is feasible to $P_{n-1}[z^{n-1}(x_n,z)]$. It follows from Assumption 9.2 that there exists some $(x'_1,\ldots,x'_{n-1}) \in E_{n-1}[z^{n-1}(x_n,z)]$ satisfying

$$F_\ell^{n-1}(f_{\ell 1}(x'_1),\ldots,f_{\ell n-1}(x'_{n-1}))$$

$$\geqq F_\ell^{n-1}(f_{\ell 1}(x_1),\ldots,f_{\ell n-1}(x_{n-1})) \quad \text{for all } \ell . \tag{9.9}$$

By (9.2), (9.9) and the monotonicity of ϕ_ℓ^n , we have

$$F_\ell^n(f_{\ell 1}(x'_1),\ldots,f_{\ell n-1}(x'_{n-1}),f_{\ell n}(x_n))$$

$$\geqq F_\ell^n(f_{\ell 1}(x_1),\ldots,f_{\ell n-1}(x_n)) \quad \text{for all } \ell . \tag{9.10}$$

Since $(x'_1,\ldots,x'_{n-1},x_n)$ is feasible to (9.7), it follows from (9.8) and (9.10) that $(x_1^o,\ldots,x_n^o)$ cannot be an efficient solution of (9.7). This contradicts the assumption and completes the proof. □

Clearly if $(x_1^o,\ldots,x_n^o)$ is efficient to $P_n[z]$, then $(x_1^o,\ldots,x_n^o)$ is also efficient to (9.7), because, from Theorem 9.1, $(x_1^o,\ldots,x_n^o)$ is feasible to problem (9.7) and the feasible set of (9.7) is contained in that of $P_n[z]$.

Consequently, from Theorems 9.1 and 9.2, we obtain the following result which is a generalization of the recursive functional equation of dynamic programming for ordinary mathematical programming problems given in Chapter 7 to the case of multiple criteria. A similar equation is given by Klötzler [K4] for a multistage decision process.

Theorem 9.3. Let Assumptions 9.1 and 9.2 be satisfied. Then the following recursive relations hold for $n = 2, \ldots, N$,

$$\mathcal{F}^n[z] = \underset{x_n \in X_n}{\text{Maximal}} \; \phi^n(\mathcal{F}^{n-1}[z^{n-1}(x_n, z)], f_n(x_n)) \quad ,$$

where $\mathcal{F}^n[z]$ is the image of the set $E_n[z]$ under the mapping $F^n(f_1, \ldots, f_n)$ and the right-hand-side is understood to be the set of maximal points of the set $\{ y \in R^L \; ; \; y = \phi^n(v, f_n(x_n)) \; , v \in \mathcal{F}^{n-1}[z^{n-1}(x_n, z)] \; , \; x_n \in X_n \}$.

Proof. From the discussion above, the set $E_n[z]$ coincides with the set of efficient solutions to problem (9.7). Hence the theorem immediately follows. $\square$

Next we state a theorem concerning the properly efficiency of subproblems. To this end, we need an assumption on the functions ϕ^n.

Assumption 9.3. For any $n = 2, \ldots, N$, the functions ϕ^n_ℓ satisfy the following property: There exist scalars $a_n > 0$ and A_n such that

$$a_n \, | \, \xi - \eta \, | \; \leqq \; | \, \phi^n_\ell(\xi, \zeta) - \phi^n_\ell(\eta, \zeta) \, | \; \leqq A_n \, | \, \xi - \eta \, | \qquad (9.11)$$

for all ξ, η, ζ and all ℓ.

This inequality (9.11) obviously holds if $\phi^n_\ell(\cdot, \zeta)$ and $1/\phi^n_\ell(\cdot, \zeta)$ are Lipschitz continuous. Note that since ϕ^n_ℓ is monotone increasing with respect to the first argument, (9.11) implies that for any ξ, η, ζ such that $\xi > \eta$.

$$0 < a_n (\xi - \eta) \leqq \phi_\ell^n(\xi,\zeta) - \phi_\ell^n(\eta,\zeta) \leqq A_n (\xi - \eta) . \tag{9.12}$$

The following theorem states the relationship between the set of properly efficient solutions of the nth subproblem and that of the (n-1)th subproblem.

Theorem 9.4. Suppose that Assumptions 9.1, 9.2, and 9.3 are satisfied. If for each n , $n = 2,\dots,N$, $(x_1^o,\dots,x_n^o)$ is a properly efficient solution of problem $P_n[z]$, then $(x_1^o,\dots, x_{n-1}^o)$ is also a properly efficient solution of problem $P_{n-1}[z^{n-1}(x_n^o,z)]$.

Proof. Since $(x_1^o,\dots,x_n^o) \in E_{n-1}[z^{n-1}(x_n^o,z)]$ has already been proved in Theorem 9.1, we need only show that it is proper. Pick any $(x_1,\dots,x_{n-1})$ in the feasible set of problem $P_{n-1}[z^{n-1}(x_n^o,z)]$. Let k be any index such that

$$\begin{aligned} &F_k^{n-1}(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1})) \\ &\qquad > F_k^{n-1}(f_{k1}(x_1^o),\dots,f_{kn-1}(x_{n-1}^o)) . \end{aligned} \tag{9.13}$$

By (9.2), (9.13) and the monotonicity of ϕ_k^n , we have

$$\begin{aligned} &F_k^n(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1}),f_{kn}(x_n^o)) \\ &\qquad > F_k^n(f_{k1}(x_1^o),\dots,f_{kn}(x_n^o)) . \end{aligned}$$

Since $(x_1^o,\dots,x_n^o) \in PE_n[z]$ and $(x_1,\dots,x_{n-1},x_n^o)$ is feasible to problem $P_n[z]$, for some scalar $\mu > 0$ and some index ℓ such that

$$F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1}),f_{\ell n}(x_n^o)) < F_\ell^n(f_{\ell 1}(x_1^o),\dots,f_{\ell n}(x_n^o)) ,$$

we must have

$$\frac{F_k^n(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1}),f_{kn}(x_n^o)) - F_k^n(f_{k1}(x_1^o),\dots,f_{kn}(x_n^o))}{F_\ell^n(f_{\ell 1}(x_1^o),\dots,f_{\ell n}(x_n^o)) - F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1}),f_{\ell n}(x_n^o))} \leqq \mu . \qquad (9.14)$$

From (9.2), (9.12), and (9.13), it is seen that

$$\begin{aligned}&\{\text{the numerator of (9.14)}\}\\ &\geqq a_n\{F_k^{n-1}(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1}))\\ &\qquad - F_k^{n-1}(f_{k1}(x_1^o),\dots,f_{kn-1}(x_{n-1}^o))\}\end{aligned}$$

and

$$\begin{aligned}&\{\text{the denominator of (9.14)}\}\\ &\leqq A_n\{F_\ell^{n-1}(f_{\ell 1}(x_1^o),\dots,f_{\ell n-1}(x_{n-1}^o))\\ &\qquad - F_\ell^{n-1}(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1}))\} .\end{aligned}$$

Thus it follows that

$$\frac{F_k^{n-1}(f_{k1}(x_1),\dots,f_{kn-1}(x_{n-1})) - F_k^{n-1}(f_{k1}(x_1^o),\dots,f_{kn-1}(x_{n-1}^o))}{F_\ell^{n-1}(f_{\ell 1}(x_1^o),\dots,f_{\ell n-1}(x_{n-1}^o)) - F_\ell^{n-1}(f_{\ell 1}(x_1),\dots,f_{\ell n-1}(x_{n-1}))} \leqq A_n\mu/a_n .$$

This shows that $(x_1^o,\ldots,x_{n-1}^o) \in PE_{n-1}[z^{n-1}(x_n^o,z)]$ and completes the proof. $\square$

In this section, we have established several recursive relations to solve multiple criteria mathematical programming problems by dynamic programming. Computational aspects of these methods will be discussed in the next section.

9.4 Computational Aspects

In this section, we discuss some computational aspects for finding efficient and properly efficient solutions of multiple criteria mathematical programming problems. Multiple criteria problems are frequently reduced into scalar maximization problems by introducing certain parameters. Perhaps, the most intuitive scalar-criterion mathematical programming problem reduced from the original multiple criteria problem is the following:

$$
\begin{array}{lll}
\text{maximize} & F_1(f_{11}(x_1),\dots,f_{1N}(x_N)) & \\
\text{subject to} & F_\ell(f_{\ell 1}(x_1),\dots,f_{\ell N}(x_N)) \leqq b\ , & \ell=2,\dots,L\ , \\
 & G_m(g_{m1}(x_1),\dots,g_{mN}(x_N)) \leqq 0\ , & m=1,\dots,M\ , \\
\text{and} & x_n \in X_n\ , & n=1,\dots,N\ ,
\end{array}
\tag{9.15}
$$

where $b = (b_2,\dots,b_L)$ is an $(L-1)$-vector such that (9.15) is not infeasible. It is clear from the definition of efficiency that, for each fixed b, at least one optimal solution of problem (9.15) is an efficient solution of problem (9.1). If Assumption 9.1 is satisfied, problem (9.15) can be decomposed into subproblems by dynamic programming in the sense of Chapter 7, since the conditions therein are satisfied by problem (9.15). In this case, defining the real valued functions $p_n(\varepsilon,z)$ for each $n=1,\dots,N$, each $\varepsilon = (\varepsilon_2,\dots,\varepsilon_L)$, and each $z = (z_1,\dots,z_M)$ by

$$
\begin{aligned}
p_n(\varepsilon,z) = \max \{\ & F_1^n(f_{11}(x_1),\dots,f_{1n}(x_n)) \ | \\
& F_\ell^n(f_{\ell 1}(x_1),\dots,f_{\ell n}(x_n)) \leqq \varepsilon_\ell,\ \ell=2,\dots,L,
\end{aligned}
$$

$$G_m^n(g_{m1}(x_1),\dots,g_{mn}(x_n)) \leqq z_m \ , \ m=1,\dots,M \ ,$$

$$\text{and} \quad x_1 \in X_1 \ ,\dots, \ x_n \in X_n \ \} \ ,$$

we obtain recursive relations of the form

$$p_n(\varepsilon,z) = \max_{x_n \in X_n} \ \phi_1^n(p_n(\varepsilon^{n-1}(x_n,\varepsilon),z^{n-1}(x_n,z)),f_{1n}(x_n)) \ , \qquad (9.16)$$

for $n=2,\dots,N$, where $\varepsilon^{n-1}(\cdot,\cdot)$ is defined for F_ℓ^n in a manner quite similar to (9.4). Obviously, problem (9.15) is identical with problem of finding $P_N(0,b)$. Hence, generating efficient solutions of problem (9.1) reduces to solving (9.16) recursively to obtain $P_N(b,0)$ for various values of b . It is noted that any F_ℓ can be chosen as an objective function in place of F_1 in problem (9.15).

Another way of reduction of multiple criteria problems into scalar maximization problems which give efficient solutions is a scalarization method of the following type:

$$\text{maximize} \quad \sum_{\ell=1}^{L} \lambda_\ell F_\ell(f_{\ell 1}(x_1),\dots,f_{\ell N}(x_N)) \qquad (9.17)$$

$$\text{subject to} \quad G_m(g_{m1}(x_1),\dots,g_{mN}(x_N)) \leqq 0 \ , \ m=1,\dots,M \ ,$$

$$\text{and} \qquad x_n \in X_n \ , \qquad n=1,\dots,N \ ,$$

where $\lambda = (\lambda_1,\dots,\lambda_L)$ is an L-vector with positive components. It is known [G2] that if $(x_1^o,\dots,x_N^o)$ is an optimal solution of (9.17), then $(x_1^o,\dots,x_N^o)$ is also a properly efficient solution of problem (9.1). Conversely, every properly efficient solution of problem (9.1) is an optimal solution of (9.17) for some $\lambda = (\lambda_1,\ldots,\lambda_L) > 0$, provided that every X_n is closed and convex, and F_ℓ and G_m are concave and convex, respectively, in $(x_1,\ldots,x_N)$. The latter is particularly true if F_ℓ and G_m are concave and convex, respectively, and are increasing with respect to $(f_{\ell 1},\ldots, f_{\ell N})$ and $(g_{m1},\ldots,g_{mN})$, respectively, and if $f_{\ell n}$ and g_{mn} are concave and convex, respectively, in $(x_1,\ldots,x_N)$. Moreover, when each X_n is a linear polytope and each F_ℓ is linear, every efficient solution of problem (9.1) solves (9.17) for some $\lambda = (\lambda_1,\ldots, \lambda_L) > 0$, since the set of efficient solutions is identical with that of proper efficient solutions [B5].

Let us suppose that every F_ℓ is additive, i.e., $\ell = 1,\ldots,L$,

$$F_\ell(f_{\ell 1}(x_1),\ldots,f_{\ell N}(x_N)) = f_{\ell 1}(x_1) + \ldots + f_{\ell N}(x_N) .$$

Then the objective function in (9.17) becomes $\sum_{n=1}^{N} \sum_{\ell=1}^{L} \lambda_\ell f_{\ell n}(x_n)$ or $\sum_{n=1}^{N} \lambda f_n(x_n)$. If we define real valued functions $q_n(\lambda,z)$ for $n = 1,\ldots,N$, each $\lambda = (\lambda_1,\ldots,\lambda_L) > 0$, and each $z = (z_1,\ldots,z_M)$ by

$$\begin{aligned} q_n(\lambda,z) = \max \{ \sum_{i=1}^{n} \lambda f_i(x_i) \mid \\ & G_m^n(g_{m1}(x_1),\ldots,g_{mn}(x_n)) \leq z_m , \ m = 1,\ldots,M , \\ & \text{and } x_1 \in X_1 ,\ldots, x_n \in X_n \} , \end{aligned}$$

we can obtain the recursive relations for $n = 2,\ldots,N$,

$$q_n(\lambda,z) = \max_{x_n \in X_n} \{ q_{n-1}(\lambda,z^{n-1}(x_n,z)) + \lambda f_n(x_n) \} . \tag{9.18}$$

Using the recursive relations (9.18) for various values of λ, we may find a set of properly efficient solutions of problem (9.1) by obtaining $q_N(\lambda,0)$.

Although we have stated the recursive relations (9.16) and (9.18) without proof, the validity of those relations is seen from the results obtained in Chapter 7.

Finally, we compare problems (9.15) and (9.17) or formulas (9.16) and (9.18). An advantage of (9.15) over (9.17) is in that (9.15) can treat nonconvex problems as well as convex problems, whereas (9.17) may fail to find some efficient solutions when used for nonconvex problems. On the other hand, the merit of (9.17) is a small number of constraints relative to (9.15). Since the number of constraints is exactly the dimensionality of state space in the dynamic programming formulation and computational effort is seriously affected by the dimensionality of state space, so called "curse of dimensionality", we may expect that cost of solving (9.18) is considerably smaller than that of solving (9.16).

9.5 Conclusion

We have described a dynamic programming approach to multiple criteria mathematical programming problems. A remarkable characteristic of the recursive functional equations derived here is that the relations are given between the maximal set of multiple criteria of the nth subproblem and that of the (n-1)th subproblem. In fact, this is the outstanding feature that distinguishes the present method from that of single criterion problems, for which the relations are given between the maximum value of the nth subproblem and that of the (n-1)th subproblem.

It is noted that dynamic programming has been applied to non-scalar criterion problems other than mathematical programming problems. For example, Mitten [M17] and Sobel [S4] obtained some results for finite and infinite horizon sequential decision processes, and Klötzler [K4] reported a result on multistage optimal decision problems.

To multiple criteria problems, there are other approaches that make explicit or implicit use of utility functions [F9]. Such methods, which are sometimes interactive, enable us to find some preferable solution among the set of efficient solutions, and thus are rather relevant to actual decision making problems. To study the application of dynamic programming from such a viewpoint is one of the important subjects for future reseacrh.

CHAPTER 10

CONCLUSION

Throughout the thesis, we have considered nonlinear programming problems from a variety of viewpoints. The thesis is roughly divided into three parts. (i) Unification of penalty function methods has been examined for convex programs. Furthermore, the dual approaches have been proposed to overcome computational difficulties inherent to these methods. The key to the dualization is Fenchel's duality theorem in convex analysis. (ii) A certain type of nonconvex programs have been considered and two algorithms for solving these problems have been proposed. The first algorithm, regarded as a natural extension of the Frank-Wolfe method, is applicable to considerably general problems. The second one is a generalization of the proximal point algorithm for convex problems to deal with nonconvex functions. The two proposed algorithms are closely related to each other in the sense that both of them convexify the problem by linearizing the nonconvex term and thus make the theory of convex analysis applicable. (iii) Dynamic programming has been applied in a unified manner to a variety of problems such as large-scale nonlinear programming problems, nonlinear chance-constrained programming problems and multiple criteria mathematical programming problems. Sufficient conditions have been given for each of these problems to be decomposed and to be solved via recursive functional equations of dynamic programming.

A wide variety of mathematically modeled actual problems can be formulated as nonlinear programming problems. Nonlinear programming will undoubtedly increase its importance as a practical tool in the field of engineering, and the developments in both theory and applications will continue to enhance the future research. The author hopes that the work contained in this thesis will be of help in moving the status of nonlinear programming one step forward.

APPENDIX

CONVEX SETS AND FUNCTIONS

In this appendix we summarize some definitions and notations in convex analysis that are needed in the thesis. The reader should refer to Rockafellar [R5] for the details. Some of them can also be found elsewhere [F2][L7][R7][S6].

Let C be a set in R^n. C is said to be *convex* if $(1-\lambda)x + \lambda y \in C$ whenever $x \in C$, $y \in C$ and $0 < \lambda < 1$. The *relative interior* of C is defined as the interior of C with respect to the smallest affine set containing C and is denoted by ri C . The interior of C in the ordinary sense is denoted by int C .

All functions are understood to be extended-real-valued. A *proper convex function* f is a function, whose values are in $(-\infty,+\infty]$ and not identically $+\infty$, such that $f(\lambda x+(1-\lambda)y) \leqq \lambda f(x) + (1-\lambda)f(y)$ for all x, y and $0 < \lambda < 1$. A *proper concave function* g is a function such that $-g$ is a proper convex function. In what follows, we omit the term "proper" since improper functions appear nowhere in the thesis.

For a convex function f and a concave function g on R^n, the sets epi $f = \{ (x,\mu) \in R^{n+1}; \mu \geqq f(x) \}$ and epi $g = \{ (x,\mu) \in R^{n+1}; \mu \leqq g(x) \}$ are called the *epigraphs* of f and g , respectively. Moreover, the sets dom $f = \{ x \in R^n; f(x) < +\infty \}$ and dom $g = \{ x \in R^n; g(x) > +\infty \}$ are called the *effective domains* of f and g , respectively. A convex (concave) function is said to be *closed* if its epigraph is a closed set.

Norms are convex functions of particular importance. The ℓ_p norm of a vector $x = (x_1, \ldots, x_n)^T$ in R^n (T denotes the transposition) is defined as follows:

$$\| x \|_p = \begin{cases} (\sum_{i=1}^{m} |x_i|^p)^{1/p} & 1 \leqq p < \infty \\ \max \{|x_1|, \ldots, |x_n|\} & p = \infty \end{cases} .$$

The ℓ_1 norm and ℓ_2 norm are often called the rectangular norm and the Euclidean norm, respectively. Normally, the Euclidean norm is also denoed by $\| \cdot \|$. It is noted that any norm is a *gauge*, i.e., a nonnegative positively homogeneous convex function with value zero at the origin.

Conjugacy of convex or concave functions is one of the striking topics in convex analysis. The basis of conjugacy is the fact that the epigraph of a convex (concave) function is the intersection of the closed halfspaces that contain it. The *conjugate* of a convex function f on R^n is a function f^* on R^n defined by

$$f^*(y) = \sup_{x \in R^n} \{ \langle x,y \rangle - f(x) \} ,$$

where $\langle \cdot , \cdot \rangle$ denotes the inner product. The f^* is closed and convex. Moreover, if f is closed, the conjugate of f^* is again f . Therefore the conjugacy correspondence is one-to-one in the class of all closed convex functions. Similarly, the conjugate of a concave function g on R^n is a closed concave function g^* on R^n defined by

$$g^*(y) = \inf_{x \in R^n} \{ \langle x,y \rangle - g(x) \} .$$

Let ϕ be a nondecreasing closed convex function on $[0,+\infty)$ with $\phi(0)$ finite. The *monotone conjugate* of ϕ, which we denote ϕ^+, is defined by

$$\phi^+(\eta) = \sup \{ \xi\eta - \phi(\xi) \mid \xi > 0 \} .$$

It should be noted that the monotone conjugacy defines a symmetric one-to-one correspondence in the class of all nondecreasing closed convex functions on $[0,+\infty)$ which are finite at zero. In fact, the monotone conjugacy correspondence may be viewed as a special class of the conjugacy correspondence. In comparison with the conjugates of convex functions on R^n, the monotone conjugates are generally easier to calculate, since the domain of the functions is a real half line. Moreover, it is not unusual to obtain the monotone conjugate functions analytically.

The classical notion of differentiability of functions can be extended to the notion of subdifferentials when functions are convex or concave. The *subgradient* of a convex function f at x is a vector y such that

$$f(z) \geqq f(x) + \langle y,z-x\rangle \qquad \text{for any } z .$$

The *subdifferential* of f at x is the set of all subgradients of f at x, and is denoted by $\partial f(x)$. For a concave function g, the subgradient of g at x is a vector y such that

$$g(z) \leqq g(x) + \langle y,z-x\rangle \qquad \text{for any } z .$$

The set of all such y is also called the subdifferential of g at x, and is denoted by $\partial g(x)$. Note that subgradients of convex or concave functions correspond to supporting hyperplanes to epigraphs of functions. In particular, if functions are differentiable in the ordinary sense, then the subgradients reduce, of course, to the gradients and the usual notation such as $\nabla f(x)$ or $\nabla g(x)$ is used.

Asymptotic properties of convex sets and convex (concave) functions are important in the development of existence theorems in convex analysis. The notion of 'recession' will be useful in formulating various growth conditions that specify some behavior of sets and functions at infinity. A *direction of recession* of a convex set C is a nonzero vector y such that $x + \lambda y \in C$ for every $\lambda \geqq 0$ and $x \in C$. The *recession cone* of C, denoted by 0^+C, is the set consisting of the zero vector and all directions of recession of C.

The *recession function* of a convex function f is defined as a function whose epigraph is the recession cone of $\text{epi } f$ in R^{n+1}. We denote it by $f0^+$. Then $\text{epi } f0^+ = 0^+(\text{epi } f)$ by definition. The recession function $f0^+$ of f is a positively homogeneous, i.e., $(f0^+)(\lambda y) = \lambda(f0^+)(y)$ for every $\lambda > 0$, convex function. One has

$$(f0^+)(y) = \lim_{\lambda \downarrow 0} (f\lambda)(y) \qquad \text{for every } y \in \text{dom } f ,$$

where $(f\lambda)(y) = \lambda f(y/\lambda)$. The set of all vectors y such that $(f0^+)(y) \leqq 0$ is called the *recession cone* of f and such vectors are called the *directions of recession* of f. For a concave function g, the recession function $g0^+$ can be defined in a similar manner. Directions of recession of g are vectors y such that $(g0^+)(y) \geqq 0$ and the set of those vectors is the recession cone of g .

Although we make no distinction about symbols, e.g., $*$, ∂, 0^+, between for convex functions and for concave functions, the meanings will be clear from context.

A fundamental and beautiful duality theorem proved by Fenchel [F2] is one of the splendid results in convex analysis. This theorem has been refined by Rockafellar [R3] in a generalized form. These theorems play a central role in the development of Chapters 3 and 4 of this thesis. We now state these theorems without proof. Complete proofs may be found in [R5].

Fenchel's Duality Theorem Let f and g be a convex function and a concave function on R^n, respectively. If $\mathrm{ri}(\mathrm{dom}\ f) \cap \mathrm{ri}(\mathrm{dom}\ g) \neq \emptyset$, then

$$\inf_{x \in R^n} \{ f(x) - g(x) \} = \sup_{y \in R^n} \{ g^*(y) - f^*(y) \} ,$$

where the supremum is attained at some y .

Fenchel's Duality Theorem (refined by Rockafellar) Let f be a closed convex function on R^n , let g be a closed concave function on R^m , and let A be a linear transformation from R^n to R^m . If there exists an $x \in \mathrm{ri}(\mathrm{dom}\ f)$ such that $Ax \in \mathrm{ri}(\mathrm{dom}\ g)$, then

$$\inf_{x \in R^n} \{ f(x) - g(Ax) \} = \sup_{y \in R^m} \{ g^*(y) - f^*(A^T y) \} ,$$

where the supremum is attained at some y .

REFERENCES

[A1] J. Abadie, Application of the GRG algorithm to optimal control problems, *Integer and Nonlinear Programming,* J. Abadie, ed., North-Holland, Amsterdam, The Netherlands, 1970, pp. 191-211.

[A2] ________ and J. Carpentier, Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, *Optimization*, R. Fletcher, ed., Academic Press, London, 1969, pp. 37-47.

[A3] R. A. Abrams and A. Ben-Israel, Optimality conditions and recession cones, *Operations Res.*, 23 (1975), pp. 549-553.

[A4] R. R. Allran and S. E. J. Johnsen, An algorithm for solving nonlinear programming problems subject to nonlinear inequality constraints, *Computer J.*, 13 (1970), pp. 171-177.

[A5] L. Armijo, Minimization of functions having Lipschitz-continuous first partial derivatives, *Pacific J. Math.*, 16 (1966), pp. 1-3.

[A6] K. J. Arrow, F. J. Gould and S. M. Howe, A general saddle point result for constrained optimization, *Math. Programming*, 5 (1973), pp. 225-234.

[A7] ___________ and R. M. Solow, Gradient methods for constrained maxima, with weakened assumptions, *Studies in Linear and Nonlinear Programming*, K. J. Arrow, L. Hurwicz and H. Uzawa, eds., Stanford Univ. Press, Stanford, California, 1958, pp. 166-176.

[B1] M. S. Bazaraa, J. J. Goode and C. M. Shetty, Constraint qualification revisited, *Management Sci.*, 18 (1972), pp. 567-573.

[B2] R. Bellman, *Dynamic Programming*, Princeton Univ. Press, Princeton, N. J., 1957.

[B3] ________ and S. Dreyfus, *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, N. J., 1962.

[B4] J. F. Benders, Partitioning procedures for solving mixed-variables programming problems, *Numer. Math.*, 4 (1962), pp. 238-252.

[B5] H. P. Benson and T. L. Morin, The vector maximization problem: Proper efficiency and stability, *SIAM J. Appl. Math.*, 32 (1977), pp. 64-72.

[B6] C. Berge, *Topological Spaces*, Oliver & Boyd, Edinburgh and London, 1963.

[B7] D. P. Bertsekas, Combined primal-dual and penalty methods for constrained minimization, *SIAM J. Control*, 13 (1975), pp. 521-544.

[B8] ______________, Necessary and sufficient conditions for a penalty function to be exact, *Math. Programming*, 9 (1975), pp. 87-99.

[B9] ______________, Multiplier methods: A survey, *Automatica*, 12 (1976), pp. 133-145.

[B10] ______________, On penalty and multiplier methods for constrained minimization, *SIAM J. Control*, 14 (1976), pp. 216-235.

[B11] ______________ and S. K. Mitter, A descent numerical method for optimization problems with nondifferentiable cost functionals, *SIAM J. Control*, 11 (1973), pp. 637-652.

[C1] A. Charnes and W. W. Cooper, Deterministic equivalents for optimizing and satisficing under chance-constraints, *Operations Res.*, 11 (1963), pp. 18-39.

[C2] __________, M. J. L. Kirby and W. M. Raike, An acceptance region theory for chance-constrained programming, *J. Math. Anal. Appl.*, 32 (1970), pp. 38-61.

[C3] F. K. Clarke, Generalized gradients and applications, *Trans. Amer. Math. Soc.*, 205 (1975), pp. 247-262.

[C4] __________, A new approach to Lagrange multipliers, *Math. of Operations Res.*, 1 (1976), pp. 165-174.

[C5] J. L. Cochrane and M. Zeleny (eds.), *Multiple Criteria Decision Making*, Univ. of South Carolina Press, Columbia, S. C., 1973.

[C6] A. W. Conn, Constrained optimization using a nondifferentiable penalty function, *SIAM J. Numer. Anal.*, 10 (1973), pp. 760-784.

[D1] G. B. Dantzig and P. Wolfe, Decomposition principle for linear programs, *Operations Res.*, 8 (1960), pp. 101-111.

[D2] __________ and ________, The decomposition algorithm for linear programs, *Econometrica*, 29 (1961), pp. 767-778.

[E1] J. P. Evans and F. J. Gould, Stability and exponential penalty function techniques in nonlinear programming, Institute of Statistics Mimeo Series, No. 723, Univ. of North Carolina, Chapel Hill, 1970.

[E2] ____________ and ___________, An existence theorem for penalty function theory, *SIAM J. Control*, 12 (1974), pp. 509-516.

[E3] ____________, ___________ and J. W. Tolle, Exact penalty functions in nonlinear programming, *Math. Programming*, 4 (1973), pp. 72-97.

[F1] J. E. Falk, Lagrange multipliers and nonconvex programs, *SIAM J. Control*, 7 (1969), pp. 534-545.

[F2] W. Fenchel, *Convex Cones, Sets and Functions*, mimeographed lecture notes, Princeton Univ., Princeton, N. J., 1951.

[F3] A. Feuer, Minimizing well-behaved functions, *Proc. of Twelfth Annual Allerton Conf. on Circuit and System Theory*, Illinois, 1974, pp. 15-34.

[F4] ________, An extension of the method of conjugate subgradients to generalized minimax objectives, Yale Univ., New Haven, Conn., July 1976.

[F5] A. V. Fiacco, A general regularized sequential unconstrained minimization technique, *SIAM J. Appl. Math.*, 17 (1969), pp. 1239-1245.

[F6] ____________, Penalty methods for mathematical programming in E^n with general constraint sets, *J. Opt. Theory and Appl.*, 6 (1970), pp. 252-268.

[F7] ____________ and A. P. Jones, Generalized penalty methods in topological spaces, *SIAM J. Appl. Math.*, 17 (1969), pp. 996-1000.

[F8] ___________ and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, New York, 1968.

[F9] P. Fishburn, *Utility Theory for Decision Making*, John Wiley, New York, New York, 1970.

[F10] R. Fletcher, An ideal penalty function for constrained optimization, *J. Inst. Math. Appl.*, 15 (1970), pp. 319-342.

[F11] R. L. Francis and J. M. Goldstein, Location theory: A selective bobliography, *Operations Res.*, 22 (1974), pp. 400-410.

[F12] M. Frank and P. Wolfe, An algorithm for quadratic programming, *Naval Res. Logistics Quartery*, 3 (1956), pp. 95-110.

[F13] M. Fukushima, Mathematical programming for nondifferentiable functions, in Japanese, *Operations Research (OR Society of Japan)*, 23 (1978), pp. 317-324, 384-393.

[F14] ___________, Minimization methods for the sum of a convex function and a continuously differentiable function, to be presented at *the Tenth Int. Symp. on Math. Programming*, Montreal, Aug. 1979.

[G1] D. Gale, A geometric duality theorem with economic applications, *Rev. Econ. Studies*, 34 (1967), pp. 19-24.

[G2] A. M. Geoffrion, Proper efficiency and the theory of vector minimization, *J. Math. Anal. Appl.*, 22 (1968), pp. 618-630.

[G3] ______________, Elements of large-scale mathematical programming: Parts I and II, *Management Sci.*, 16 (1970), pp. 652-691.

[G4] ______________, Primal resource-directive approaches for optimizing nonlinear decomposable systems, *Operations Res.*, 18 (1970), pp. 375-403.

[G5] ______________, Duality in nonlinear programming: A simplified applications-oriented development, *SIAM Review*, 13 (1971), pp. 1-37.

[G6] ______________, Generalized Benders decomposition, *J. Opt. Theory and Appl.*, 10 (1972), pp. 237-260.

[G7] D. Goldfarb, Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints, *SIAM J. Appl. Math.*, 17 (1969), pp. 739-764.

[G8] A. A. Goldstein, Optimization of Lipschitz continuous functions, *Math. Programming*, 13 (1977), pp. 14-22.

[G9] F. J. Gould, Extensions of Lagrange multipliers in nonlinear programming, *SIAM J. Appl. Math.*, 17 (1969), pp. 1280-1297.

[G10] ___________, A class of inside-out algorithms for general programs, *Management Sci.*, 16 (1970), pp. 350-356.

[G11] H. J. Greenberg, Dynamic programming with linear uncertainty, *Operations Res.*, 16 (1968), pp. 675-678.

[H1] S. P. Han, Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Math. Programming*, 11 (1976), pp. 263-282.

[H2] M. R. Hestenes, Multiplier and gradient methods, *J. Opt. Theory and Appl.*, 4 (1969), pp. 303-320.

[H3] D. M. Himmelblau (ed.), *Decomposition of Large-Scale Problems*, North-Holland, Amsterdam, The Netherlands, 1973.

[H4] J. B. Hiriart-Urruty, On optimality conditions in nondifferentiable programming, *Math. Programming*, 14 (1978), pp. 73-86.

[H5] A. S. Housholder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, New york, 1964.

[H6] S. Howe, New conditions for exactness of a simple penalty function, *SIAM J. Control*, 11 (1973), pp. 378-381.

[J1] R. Jagannathan, Chance-constrained programming with joint constraints, *Operations Res.*, 22 (1974), pp. 358-372.

[J2] F. John, Extremum problems with inequalities as subsidiary conditions, *Studies and Essays, Courant Anniversary Volume*, Interscience, New York, New York, 1948, pp. 187-204.

[J3] H. Juel and R. Love, On the dual of the linearly constrained multifacility location problems with arbitrary norms, MRC Tech. Summary Report #1721, Math. Res. Center, Univ. of Wisconsin-Madison, Madison, WI, Feb. 1977.

[K1] R. M. Karp, Functional decomposition and switching circuit design, *J. Soc. Indust. Appl. Math.*, 11 (1963), pp. 291-335.

[K2] W. Karush, Minima of functions of several variables with inequalities as side conditions, M. S. Dissertation, Dept. of Math., Univ. of Chicago, Chicago, Illinois, Dec. 1939.

[K3] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley, New York, New York, 1976.

[K4] R. Klötzler, Multiobjective dynamic programming, Paper presented at *the Ninth Int. Symp. on Math. Programming*, Budapest, Aug. 1976.

[K5] B. W. Kort and D. P. Bertsekas, Combined primal-dual and penalty methods for convex programming, *SIAM J. Control and Opt.*, 14 (1976), pp. 268-294.

[K6] H. W. Kuhn and A. W. Tucker, Nonlinear programming, *Proc. of the Second Berkeley Symp. on Math. Stat. and Probability*, J. Neyman, ed., Univ. of California Press, California, 1951, pp. 481-492.

[L1] L. S. Lasdon, *Optimization Theory for Large Systems*, MacMillan, New York, New York, 1970.

[L2] C. Lemarechal, An extension of Davidon methods to nondifferentiable problems, *Math. Programming Study*, No. 3 (1975), pp. 95-109.

[L3] E. S. Levitin and B. T. Polyak, Constrained minimization methods, *USSR Comput. Math. Math. Physics*, 6(5) (1966), pp. 1-50.

[L4] J. G. Lin, Maximal vectors and multi-objective optimization, *J. Opt. Theory and Appl.*, 18 (1976), pp. 41-64.

[L5] F. A. Lootsma, A Survey of methods for solving constrained minimization problems via unconstrained minimization, *Numerical Methods for Nonlinear Optimization*, F. A. Lootsma, ed., Academic Press, New York, New York, 1972, pp. 313-347.

[L6] R. F. Love, The dual of a hyperbolic approximation to the generalized constrained multi-facility location problem with ℓ_p distances, *Management Sci.*, 21 (1974), pp. 22-33.

[L7] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley, New York, New York, 1969.

[L8] ________________, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[M1] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, New York, 1969.

[M2] ________________, Unconstrained Lagrangeans in nonlinear programming, *SIAM J. Control*, 13 (1975), pp. 772-791.

[M3] M. D. Mesarovic, D. Macko and Y. Takahara, *Theory of Hierarchical, Multi-level Systems*, Academic Press, New York, New York, 1970.

[M4] R. Mifflin, An algorithm for constrained optimization with semismooth functions, *Math. of Operations Res.*, 2 (1977), pp. 191-207.

[M5] __________, Semismooth and semiconvex functions in constrained optimization, *SIAM J. Control and Opt.*, 15 (1977), pp. 959-972.

[M6] H. Mine and M. Fukushima, Application of Fenchel's duality theorem to penalty methods in convex programming, *Memoirs of the Fac. of Eng.*, Kyoto Univ., Kyoto, 40 (1978), pp. 30-39.

[M7] _______ and ____________, Penalty function theory for general convex programming problems, *J. Opt. Theory and Appl.*, 24 (1978), pp. 287-301.

[M8] ______ and __________, A minimization method for the sum of a convex function and a continuously differentiable function, *J. Opt. Theory and Appl.*, to appear.

[M9] ______ and __________, Decomposition of multiple criteria mathematical programming problems by dynamic programming, *Int. J. Systems Sci.*, to appear.

[M10] ______, __________ and Y. Fujii, A dual approach to the generalized multifacility location problem with arbitrary norms, *Proc. of the Int. Conf. on Cybernetics and Society*, Tokyo, Nov. 1978, pp. 678-683.

[M11] ______, __________ and Y. J. Ryang, Parametric nonlinear programming for general cases and its application to some problems, *Memoirs of the Fac. of Eng.*, Kyoto Univ., Kyoto, 40 (1978), pp. 198-211.

[M12] ______ and K. Ohno, Decomposition of mathematical programming problems by dynamic programming and its application to block-diagonal geometric programs, *J. Math. Anal. Appl.*, 32 (1970), pp. 370-385.

[M13] ______, ______ and M. Fukushima, Multilevel decomposition of nonlinear programming problems by dynamic programming, *J. Math. Anal. Appl.*, 53 (1976), pp. 7-27.

[M14] ______, ______ and __________, Decomposition of nonlinear chance constrained programming problems by dynamic programming, *J. Math. Anal. Appl.*, 56 (1976), pp. 211-222.

[M15] ______, ______ and ___________, A 'conjugate' interior penalty method for certain convex programs, *SIAM J. Control and Opt.*, 15 (1977), pp. 747-755.

[M16] L. G. Mitten, Composition principle for systhesis of optimal multistage processes, *Operations Res.*, 12 (1964), pp. 610-619.

[M17] ____________, Preference order dynamic programming, *Management Sci.*, 21 (1974), pp. 43-46.

[M18] F. H. Murphy, A class of exponential penalty functions, *SIAM J. Control*, 12 (1974), pp. 679-687.

[M19] B. A. Murtagh and R. W. H. Sargent, A constrained minimization method with quadratic convergence, *Optimization*, R. Fletcher, ed., Academic Press, London, 1969, pp. 215-246.

[N1] G. L. Nemhauser, *Introduction to Dynamic Programming,* John Wiley, New York, New York, 1966.

[O1] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, New York, 1970.

[P1] T. Pietrzykowski, An exact potential method for constrained maxima, *SIAM J. Numer. Anal.*, 6 (1969), pp. 299-304.

[P2] A. Planchart and A. P. Hurter, Jr., An efficient algorithm for the solution of the Weber problem with mixed norms, *SIAM J. Control*, 13 (1975), pp. 650-665.

[P3] E. Polak, *Computation Methods in Optimization*, Academic Press, New York, New York, 1971.

[P4] B. T. Polyak, A general method for solving extremal problems, *Soviet Math. Doklady*, 8 (1967), pp. 593-597.

[P5] ___________, Minimization of unsmooth functionals, *USSR Comput. Math. Math. Physics*, 9(3) (1969), pp. 14-29.

[P6] M. J. D. Powell, A method for nonlinear constraints in minimization problems, *Optimization*, R. Fletcher, ed., Academic Press, London, 1969, pp. 283-298.

[P7] ______________, A fast algorithm for nonlinearly constrained optimization calculations, Paper presented at *the 1977 Dundee Conf. on Numer. Anal.*, June 1977.

[P8] B. N. Pshenichnyi, *Necessary Conditions for an Extremum*, Marcel Dekker, New York, New York, 1971.

[R1] R. T. Rockafellar, Helly's theorem and minima of convex functions, *Duke Math. J.*, 32 (1965), pp. 381-397.

[R2] ________________, Characterization of the subdifferentials of convex functions, *Pacific J. Math.*, 17 (1966), pp. 497-510.

[R3] ________________, Extension of Fenchel's duality theorem for convex functions, *Duke Math. J.*, 33 (1966), pp. 81-89.

[R4] ________________, Duality in nonlinear programming, *Mathematics of the Decision Sciences, Part I*, G. B. Dantzig and A. F. Veinott, eds., American Mathematical Society, Providence, Rhode Island, 1968, pp. 401-422.

[R5] ________________, *Convex Analysis*, Princeton Univ. Press, Princeton, N. J., 1970.

[R6] ________________, Augmented Lagrange multiplier functions and duality in nonconvex programming, *SIAM J. Control*, 12 (1974), pp. 268-285.

[R7] ________________, *Conjugate Duality and Optimization*, Regional Conference Series No. 16, Society for Industrial and Applied Mathematics, Philadelphia, Penn., 1974.

[R8] ________________, Monotone operators and the proximal point algorithm, *SIAM J. Control and Opt.*, 14 (1976), pp. 877-898.

[R9] ________________, Lagrange multipliers in optimization, *SIAM-AMS Proceedings*, Vol. 9, American Mathematical Society, Providence, Rhode Island, 1976, pp. 145-168.

[R10] ________________, Solving a nonlinear programming problem by way of a dual problem, *Symposia Mathematica*, 19 (1976), pp. 135-160.

[R11] ________________, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, *Math. of Operations Res.*, 1 (1976), pp. 97-116.

[R12] J. B. Rosen, The gradient projection method for nonlinear programming, I. Linear constraints, *J. Soc. Indust. Appl. Math.*, 8 (1960), pp. 181-217, II. Nonlinear constraints, *J. Soc. Indust. Appl. Math.*, 9 (1961), pp. 514-532.

[R13] __________ and J. C. Ornea, Solution of nonlinear programming problems by partitioning, *Management Sci.*, 10 (1963), pp. 160 -173.

[R14] D. M. Ryan, Penalty and barrier functions, *Numerical Methods for Constrained Optimization*, P. E. Gill and W. Murray, eds., Academic Press, London, 1974, pp. 175-190.

[S1] J. K. Sengupta, *Stochastic Programming, Methods and Applications,* North-Holland, Amsterdam, The Netherlands, 1972.

[S2] N. Z. Shor, Utilization of the operation of space dilatation in the minimization of convex functions, *Cybernetics*, 6 (1970), pp. 7-15.

[S3] G. J. Silverman, Primal decomposition of mathematical programs by resource allocation, *Operations Res.*, 20 (1972), pp. 58-93.

[S4] M. J. Sobel, Ordinal dynamic programming, *Management Sci.*, 21 (1975), pp. 967-975.

[S5] D. A. Sprecher, A representation theorem for continuous functions of several variables, *Proc. Amer. Math. Soc.*, 16 (1965), pp. 200-203.

[S6] J. Stoer and C. Witzgall, *Convexity and Optimization in finite Dimensions I*, Springer-Verlag, Berlin, 1970.

[T1] R. A. Tapia, Diagonalized multiplier methods and quasi-Newton methods for constrained optimization, *J. Opt. Theory and Appl.*, 22 (1977), pp. 135-194.

[V1] M. M. Vainberg, *Variational Method and Method of Monotone Operators in the Theory of Nonlinear Equations*, Israel Program for Scientific Translations Ltd., Jerusalem, Israel, 1973.

[V2] S. Vajda, *Probabilistic Programming*, Academic Press, New York, New York, 1972.

[W1] G. O. Wesolowski and R. F. Love, The optimal location of new facilities using rectangular distances, *Operations Res.*, 19 (1971), pp. 124-129.

[W2] R. A. Wijsman, Convergence of sequences of convex sets, cones and functions, *Bull. Amer. Math. Soc.*, 70 (1964), pp. 186-188.

[W3] ____________, Convergence of sequences of convex sets, cones and functions, II, *Trans. Amer. Math. Soc.*, 123 (1966), pp. 32-45.

[W4] P. Wolfe, The simplex method for quadratic programming, *Econometrica*, 27 (1959), pp. 382-398.

[W5] ________, Methods of nonlinear programming, *Nonlinear Programming*, J. Abadie, ed., North-Holland, Amsterdam, The Netherlands, 1967, pp. 97-131.

[W6] ________, A method of conjugate subgradients for minimizing nondifferentiable functions, *Math. Programming Study*, No. 3 (1975), pp. 145-173.

[Z1] W. I. Zangwill, Non-linear programming via penalty functions, *Management Sci.*, 13 (1967), pp. 344-358.

[Z2] ______________, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Englewood Cliffs, N. J., 1969.

[Z3] M. Zeleny (ed.), *Multiple Criteria Decision Making: Kyoto 1975*, Springer-Verlag, New York, New York, 1975.

[Z4] G. Zoutendijk, *Methods of Feasible Directions*, Elsevier, Amsterdam, The Netherlands, 1960.